

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
TỐ MÔN HỌC KỸ NĂNG NGHỀ NGHIỆP**



**BÁO CÁO ĐO ÁN CUỐI KỲ
MÔN VI ĐIỀU KHIỂN – VI XỬ LÝ
CHỦ ĐỀ:
ĐIỀU KHIỂN QUẠT TỰ ĐỘNG DỰA VÀO NHIỆT ĐỘ**

Nhóm thực hiện: TechStorm

Lớp: CE103.P23

Giảng viên phụ trách: TS. Đoàn Duy

Tháng 05 năm 2025

MỤC LỤC

DANH MỤC HÌNH ẢNH	V
DANH MỤC BẢNG BIÊU	VI
THÔNG TIN NHÓM THỰC HIỆN	1
QUÁ TRÌNH LÀM VIỆC NHÓM	2
CHƯƠNG 1 TỔNG QUAN VỀ ĐỀ TÀI	3
1.1 Thực trạng đề tài	3
1.2 Mục tiêu nghiên cứu	5
1.3 Giới hạn của đề tài	7
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT	10
2.1 CÁC LINH KIỆN SỬ DỤNG TRONG MẠCH	10
2.1.1 Vi điều khiển AT89S52	10
2.1.2 Cảm biến nhiệt độ LM35	19
2.1.3 Bộ chuyển đổi ADC0804	23
2.1.4 Màn hình LCD 16x02	26
2.1.5 Diode 1N4001	30
2.1.6 Fan - DC	33
2.1.7 Transistor TIP122	37
2.1.8 LED	41
2.1.9 Buzzer	44
2.2 PHẦN MỀM KEIL C	47
2.3 PHẦN MỀM PROTEUS	50
CHƯƠNG 3 THỰC HIỆN CHI TIẾT	54
3.0. Sơ đồ mạch mô phỏng trên Proteus	54
3.1. Thực hiện LM35	54
3.1.1. Vai trò của LM35	54
3.1.2. Kết nối chân của LM35	55
3.2. Thực hiện LCD 16x2	56
3.2.1. Vai trò LCD 16x2	56

3.3. Thực hiện ADC0804.....	57
3.4. Thực hiện AT89S52.....	60
3.4.1. Vai trò của vi điều khiển AT89S52 trong hệ thống của đè tài.....	61
3.4.2. Phân tích chân AT89S52 nối trong mạch	63
3.5. Quạt và Led.....	64
3.5.1. Vai trò trong hệ thống	64
3.5.2. Kết nối trong hệ thống	66
3.6. Mạch cung cấp nguồn, Reset và dao động hệ thống.....	66
3.6.1. Mạch cung cấp nguồn	66
3.6.2. Mạch cung cấp nguồn	67
3.6.3. Mạch dao động.....	67
3.7. Thiết kế phần mềm với code Assembly.....	69
3.7.1. Lưu đồ giải thuật.....	69
3.7.2 Khởi tạo cho vi xử lý	70
3.7.3 Khởi tạo và cấu hình LCD	70
3.7.6 Cấu hình Timer0	71
3.7.7 Bật ngắt toàn cục và Timer0	72
3.7.8 Hiển thị mã môn học và lớp.....	72
3.7.9 Đọc dữ liệu nhiệt độ từ cảm biến	73
3.7.10 Xử lý dữ liệu nhiệt độ	73
3.7.11 So sánh nhiệt độ và điều khiển quạt.....	74
3.7.12 Hiển thị nhiệt độ lên LCD.....	74
3.7.13 Chuyển nhiệt độ sang dạng ký tự.....	75
3.7.14 Hiển thị cảnh báo "ALERT!"	75
3.7.15 Trình phục vụ ngắt của Timer0.....	75
CHƯƠNG 4 KẾT QUẢ NGHIÊN CỨU.....	76
4.1 Kết quả nghiên cứu	76
4.2 Đánh giá hiệu năng	82
4.3 Ưu và nhược điểm của hệ thống	83

4.4. Hướng mở rộng cho hệ thống	84
4.5. Kết luận.....	85
4.6. Video mô phỏng	85
CHƯƠNG 5 SOURCE CODE VÀ TÀI LIỆU THAM KHẢO.....	86
5.1. Source code của mạch	86
5.2. Tài liệu tham khảo.....	86

DANH MỤC HÌNH ẢNH

Hình 2.1.1 a Vi điều khiển AT89S52 với các chân	10
Hình 2.1.1 b Vi điều khiển AT89S52 với các chân	11
Hình 2.1.2 Cảm biến LM35	20
Hình 2.1.3 Bộ chuyển đổi ADC0804.....	23
Hình 2.1.4 Màn hình LCD 16x2	26
Hình 2.1.5 Diode 1N4001	31
Hình 2.1.6 Fan DC	34
Hình 2.1.7 Transistor TIP122	38
Hình 2.1.8 LED	41
Hình 2.1.9 Buzzer	44
Hình 3.0 Sơ đồ mô phỏng mạch trên Proteus	54
Hình 3.7.1 Lưu đồ giải thuật	69
Hình 4.1.1 Khi nhiệt độ là 19 độ C	77
Hình 4.1.2 Khi nhiệt độ là 29 độ C	78
Hình 4.1.3 Khi nhiệt độ là 34 độ C	78
Hình 4.1.4 Khi nhiệt độ là 39 độ C	79
Hình 4.1.5 Khi nhiệt độ là 44 độ C	79
Hình 4.1.6 Khi nhiệt độ là 45 độ C(≥ 45 độ C)	80

DANH MỤC BẢNG BIỂU

Bảng 2.1.1a Sơ đồ chân VĐK AT89S52	12
Bảng 2.1.1b Mô tả chức năng các chân của AT89S52	13
Bảng 2.1.1c Các thanh ghi đặc biệt của AT89S52	15
Bảng 2.1.d Các nhóm lệnh của AT89S52	16
Bảng 2.1.e Cấu trúc hệ thống ngắt của AT89S52.....	17
Bảng 2.1.f Bảng Thanh ghi quản lí ngắt	17
Bảng 2.1.2 Cấu tạo cảm biến LM35	21
Bảng 2.1.3a Thông số kỹ thuật ADC804.....	24
Bảng 2.1.3b Thông số kỹ thuật ADC804.....	25
Bảng 2.1.4a Cấu tạo chân LCD 16x2	28
Bảng 2.1.4b Các lệnh cơ bản của LCD 16x02.....	29
Bảng 2.1.5 So sánh với các diode khác cùng dòng.....	33
Bảng 2.1.6a So sánh quạt DC và quạt AC	36
Bảng 2.1.6b Phân loại Quạt DC.....	37
Bảng 2.1.7 TIP122 có 3 chân.....	39
Bảng 2.1.8a Thông số kỹ thuật LED.....	42
Bảng 2.1.8b So sánh LED với các loại đèn truyền thông	44
Bảng 2.1.9 Thông số kỹ thuật Buzzer.....	46
Bảng 3.1 Kết nối của LM35 trong mạch.....	55
Bảng 3.2 Kết nối của LCD16x2 trong mạch.....	57
Bảng 3.3 Kết nối của ADC0804 trong mạch	60
Bảng 3.4.a So sánh AT89S52 và AT89C52	61
Bảng 3.4.b Kết nối của AT89S52 trong mạch	64
Bảng 3.5 Kết nối của Led, quạt và Buzzer trong mạch	66
Bảng 4.1 Mô phỏng mạch trong Proteus	76

THÔNG TIN NHÓM THỰC HIỆN

- Nhóm: TechStorm

- Lớp: CE103.P23

STT	Họ và tên	MSSV	Nhiệm vụ
1	Trương Thiên Quý	23521321	Thực hiện code, vẽ mô phỏng, thực hiện nối và kiểm tra mạch thực tế, làm slide, thuyết trình
2	Lê Huỳnh Thanh Phương	23521242	Thực hiện code, vẽ mô phỏng, hỗ trợ nối và kiểm tra mạch thực tế, viết báo cáo, chỉnh slide
3	Trần Đăng Quân	23521270	Viết báo cáo, hỗ trợ làm và kiểm tra mạch
4	Hồ Thanh Quang	23521280	Hỗ trợ viết báo cáo, làm slide
5	Võ Thị Ngọc Hân	23520441	Viết báo cáo, hỗ trợ làm mạch

QUÁ TRÌNH LÀM VIỆC NHÓM

- Chủ đề nghiên cứu: Điều khiển quạt tự động dựa vào nhiệt độ.
- Thời gian thực hiện: từ ngày 02/04/2025 đến ngày 28/05/2025

Thời gian họp	Nội dung	Kết quả đạt được	Hạn chế
02/04/2025	Phân công công việc	Hoàn thành Tốt	Không
17/04/2025	Thực hiện mô phỏng Proteus với code	Hoàn thành tốt	Không
27/04/2025	Tiến hành nối mạch thực tế	Phát hiện lỗi	Còn gấp một số lỗi
04/05/2025	Chỉnh sửa mạch	Hoàn thành tốt	Không
22/05/2025	Mô phỏng mạch thực tế	Hoàn thành tốt	Không
24/05/2025	Viết báo cáo và làm slide	Hoàn thành tốt	Không

CHƯƠNG 1 TỔNG QUAN VỀ ĐỀ TÀI

1.1 Thực trạng đề tài

Trong bối cảnh hiện đại, khi công nghệ đang ngày càng được tích hợp vào mọi khía cạnh của đời sống và sản xuất, nhu cầu về tự động hóa, tiết kiệm năng lượng và nâng cao hiệu suất thiết bị ngày càng trở nên cấp thiết. Một trong những khía cạnh quan trọng liên quan đến hiệu suất và độ bền thiết bị là nhiệt độ. Nhiệt độ môi trường và nhiệt độ phát sinh trong quá trình hoạt động của thiết bị đều ảnh hưởng trực tiếp đến tuổi thọ, sự ổn định và mức tiêu thụ năng lượng của hệ thống.

Các thiết bị điện – điện tử như máy tính, tủ điện, máy biến áp, thiết bị điện dân dụng, hoặc thậm chí các khu vực sinh hoạt, phòng học, phòng máy lạnh, nhà kính... đều có thể sinh nhiệt hoặc chịu tác động của nhiệt độ môi trường. Nếu không có hệ thống làm mát hoặc tản nhiệt hiệu quả, sẽ xảy ra nhiều hệ lụy nghiêm trọng như: linh kiện bị cháy, thiết bị bị ngắt tự động, giảm hiệu suất xử lý, gia tăng rủi ro cháy nổ hoặc tiêu tốn nhiều điện năng hơn để duy trì hoạt động.

Hiện nay, đa số các quạt làm mát, quạt thông gió trên thị trường vẫn sử dụng cơ chế điều khiển thủ công – người dùng phải tự bật/tắt hoặc chọn mức gió theo cảm nhận. Tuy đơn giản nhưng hình thức này bộc lộ nhiều hạn chế như:

- Không phản ứng linh hoạt với biến đổi nhiệt độ môi trường.
- Không phù hợp với các thiết bị cần làm mát tức thời theo nhiệt độ.
- Không có bảo vệ thiết bị khi bị quá nóng

Để giải quyết bài toán trên, xu hướng thiết kế hệ thống điều khiển quạt tự động theo nhiệt độ đã và đang được áp dụng rộng rãi. Các hệ thống này sử dụng cảm biến nhiệt để đo mức nhiệt độ môi trường hoặc nhiệt độ thiết bị, sau đó sử dụng vi điều khiển để xử lý tín hiệu và tự động điều chỉnh tốc độ quạt phù hợp. Đây là ứng dụng tiêu biểu của kỹ thuật tự động hóa điều khiển trong lĩnh vực điện tử – nhúng.

Khảo sát một số mô hình và công trình nghiên cứu trong và ngoài nước

Mô hình và sản phẩm thực tế

- Quạt thông minh Xiaomi Smart Fan (Trung Quốc): Tích hợp cảm biến nhiệt và điều khiển tốc độ tự động. Có khả năng điều khiển từ xa qua app, tuy nhiên giá thành cao, từ 1.5–3 triệu đồng.

- Quạt Panasonic cảm biến nhiệt độ và độ ẩm (Nhật Bản): Hoạt động tự động theo môi trường, thích hợp trong nhà tắm và phòng kín. Dù hiệu quả cao nhưng không thể tùy chỉnh theo từng yêu cầu kỹ thuật cụ thể, thiết kế phần cứng khép kín.
- Hệ thống làm mát trong máy tính: Sử dụng cảm biến kết hợp điều khiển tốc độ quạt theo xung PWM. Phổ biến trong hệ thống PC, server và laptop, tuy nhiên được tích hợp sẵn trong bo mạch chủ, khó tùy biến.

Đề tài nghiên cứu trong nước

- Luận văn tốt nghiệp – Trường ĐH Bách Khoa TP.HCM (2019): Thiết kế hệ thống làm mát tự động dùng cảm biến DS18B20 và điều khiển bằng vi điều khiển AVR. Ưu điểm là độ chính xác cao, song chương trình khá phức tạp, phần cứng khó lắp ráp thủ công.
- Đề tài sinh viên – Trường ĐH Sư phạm Kỹ thuật (2021): Ứng dụng LM35 với vi điều khiển 8051 để điều khiển quạt. Tuy đề tài gần giống, nhưng không tích hợp hiển thị LCD hoặc cảnh báo nhiệt độ cao, chưa có giao diện tương tác.
- Mô hình điều khiển quạt bằng Arduino (nhiều nguồn Youtube): Rất phổ biến trong cộng đồng maker, sử dụng LM35 hoặc DHT11 kết hợp với Arduino để điều khiển quạt. Nhược điểm: thiên về trình diễn, ít tính học thuật, không tối ưu cho vi điều khiển phổ thông như 8051.

Nhận xét và phân tích

Ưu điểm:

- Các giải pháp đều tập trung vào vấn đề thực tế: làm mát tiết kiệm và tự động theo nhiệt độ.
- Ứng dụng rộng rãi trong nhà ở, văn phòng, thiết bị điện tử, nhà kính...
- Tận dụng được các cảm biến nhiệt thông dụng (LM35, DHT11, DS18B20).

Hạn chế:

- Nhiều mô hình chỉ mang tính mô phỏng hoặc lắp ráp đơn giản.
- Phần lớn sử dụng các nền tảng cao cấp như Arduino, ESP32 – không phù hợp để đào sâu vi điều khiển nền tảng 8051.
- Chưa có mô hình tích hợp ADC rời (ADC0804), kết nối cảm biến analog với vi điều khiển AT89S52, và đồng thời điều khiển tốc độ quạt bằng PWM Timer + Ngắt – điều này giúp người học hiểu sâu hơn về điều khiển cấp thấp và cấu trúc phần cứng.

Tuy nhiên, các sản phẩm thương mại như vậy thường có giá thành cao, cấu trúc phần cứng tích hợp sẵn, khó tiếp cận để nghiên cứu – sửa chữa hoặc điều chỉnh. Đặc biệt, với đối tượng là học sinh – sinh viên hoặc các nhà nghiên cứu cá nhân, việc tìm kiếm một giải pháp thay thế có thể tự lắp ráp, dễ tùy biến và chi phí thấp là rất cần thiết.

Vì vậy, việc thực hiện đề tài “Điều khiển quạt tự động dựa vào nhiệt độ” là hoàn toàn cần thiết. Đề tài giúp sinh viên vừa tiếp cận kiến thức lý thuyết, vừa rèn luyện kỹ năng thực hành, đồng thời tạo tiền đề cho các nghiên cứu mở rộng về hệ thống điều khiển thông minh.

1.2 Mục tiêu nghiên cứu

Trong bối cảnh nhu cầu ứng dụng công nghệ tự động hóa vào các thiết bị dân dụng ngày càng cao, việc điều khiển quạt làm mát theo nhiệt độ môi trường không chỉ giúp nâng cao hiệu quả sử dụng năng lượng mà còn mang lại sự tiện lợi và an toàn cho người dùng. Đề tài “Điều khiển quạt tự động dựa vào nhiệt độ sử dụng vi điều khiển AT89S52” được thực hiện nhằm xây dựng một hệ thống đơn giản, dễ chế tạo, chi phí thấp, có khả năng hoạt động ổn định trong môi trường thực tế và phù hợp với khả năng tiếp cận của sinh viên.

Mục tiêu tổng quát:

Thiết kế và chế tạo hệ thống điều khiển quạt DC tự động theo nhiệt độ môi trường dựa trên vi điều khiển AT89S52. Hệ thống có khả năng:

- Đo nhiệt độ môi trường theo thời gian thực.
- Tự động điều chỉnh tốc độ quạt tương ứng với từng mức nhiệt.
- Hiển thị nhiệt độ lên màn hình LCD 1602.
- Cảnh báo khi nhiệt độ vượt quá ngưỡng an toàn.
- Ngắt mạch bảo vệ khi nhiệt độ quá cao.

Mục tiêu cụ thể:

1. Thiết kế phần cứng của hệ thống:

- Sử dụng cảm biến nhiệt độ LM35 để đo nhiệt độ môi trường xung quanh.
- Sử dụng ADC0804 để chuyển đổi tín hiệu tương tự từ LM35 sang tín hiệu số cho vi điều khiển.
- Dùng vi điều khiển AT89S52 để xử lý dữ liệu nhiệt độ và điều khiển hệ thống.
- Điều khiển tốc độ quạt thông qua xung PWM phát từ Timer0 + Ngắt, với transistor công suất TIP122 làm phần tử điều khiển tải.

- Tích hợp mạch cảnh báo bằng còi Buzzer khi nhiệt độ vượt quá giới hạn.
- Hiển thị giá trị nhiệt độ lên LCD1602 để người dùng dễ dàng quan sát.

2. Thiết kế phần mềm điều khiển:

- Lập trình điều khiển trên vi điều khiển AT89S52 bằng ngôn ngữ Assembly.
- Viết hàm đọc dữ liệu từ ADC0804.
- Xây dựng các mức nhiệt tương ứng với các tốc độ quạt.
- Viết chương trình hiển thị nhiệt độ lên LCD1602 theo định dạng rõ ràng.
- Tạo xung PWM để điều khiển tốc độ quạt dựa trên mức nhiệt.
- Tạo hàm cảnh báo khi vượt quá ngưỡng nhiệt độ, tự động ngắt mạch khi cần thiết.

3. Mô phỏng – thử nghiệm hệ thống:

- Thiết kế mô phỏng mạch trên Proteus.
- Mô phỏng hoạt động của cảm biến, ADC, LCD, PWM, còi và quạt.
- Thử nghiệm với các mức nhiệt độ khác nhau để đánh giá độ ổn định của hệ thống.

4. Đánh giá hiệu quả hệ thống:

- Kiểm tra tính ổn định khi thay đổi điện áp và nhiệt độ.
- Đánh giá khả năng tiết kiệm điện năng khi quạt chỉ hoạt động khi cần thiết.
- So sánh với các giải pháp tương tự về độ phức tạp, tính dễ chế tạo và chi phí.

5. Đáp ứng yêu cầu về chi phí và khả năng mở rộng:

- Sử dụng linh kiện phổ thông, dễ tìm trên thị trường, giá rẻ: AT89S52, LM35, ADC0804, TIP122, LCD1602,...
- Hệ thống có khả năng mở rộng: có thể nâng cấp để điều khiển thêm thiết bị như điều hòa, quạt hút, hoặc tích hợp IoT (nếu cần).

6. Phát triển kỹ năng cá nhân:

- Củng cố kiến thức vi điều khiển, cảm biến nhiệt, lập trình Assembly.

- Nâng cao kỹ năng vẽ mạch, mô phỏng Proteus, thiết kế và kiểm thử hệ thống thực tế.
- Hình thành kỹ năng viết báo cáo, làm việc nhóm và trình bày đề tài nghiên cứu kỹ thuật.

7. Mục tiêu học thuật và kỹ năng đạt được:

- Nâng cao kiến thức chuyên môn về vi điều khiển 8-bit, đặc biệt là AT89S52, bao gồm cấu trúc phần cứng và lập trình điều khiển.
- Thành thạo lập trình Assembly cho vi điều khiển AT89S52, sử dụng timer, ngắt, tạo PWM,...
- Rèn luyện kỹ năng sử dụng các phần mềm chuyên dụng như:
 - Proteus để mô phỏng mạch và kiểm tra tín hiệu,
 - Keil uVision để lập trình và biên dịch mã,
 - ISIS/ARES để vẽ sơ đồ nguyên lý và bố trí mạch in (nếu có).
- Củng cố kỹ năng đo lường, phân tích tín hiệu điện tử, đọc hiểu datasheet linh kiện điện tử.
- Phát triển kỹ năng viết báo cáo khoa học, trình bày logic, và thuyết trình kỹ thuật.

8. Mục tiêu nhóm (nếu đề tài thực hiện theo nhóm):

- Nâng cao kỹ năng làm việc nhóm, đặc biệt là trong môi trường kỹ thuật.
- Biết cách phân chia công việc theo năng lực từng thành viên (ví dụ: một người lập trình, một người mô phỏng, một người làm mạch thực tế, một người viết báo cáo).
- Rèn luyện kỹ năng quản lý tiến độ, xử lý xung đột, và tổng hợp kết quả nhóm.

1.3 Giới hạn của đề tài

Trong quá trình nghiên cứu và triển khai, nhóm thực hiện nhận thấy việc giới hạn phạm vi đề tài là cần thiết để đảm bảo mức độ phù hợp với năng lực sinh viên, thời gian thực hiện, cũng như cơ sở vật chất và chi phí. Cụ thể, đề tài được giới hạn ở các nội dung sau:

1.3.1. Giới hạn về phần cứng

- Vi điều khiển sử dụng:

Sử dụng vi điều khiển AT89S52 thuộc họ vi điều khiển 8051 truyền thống. Đây là loại

vi điều khiển phổ biến trong giáo dục, có số lượng chân I/O đủ để điều khiển hệ thống đơn giản. Đề tài không sử dụng các dòng vi điều khiển có tích hợp sẵn ADC hoặc các module hiện đại như ARM, Arduino, ESP8266, ESP32.

- Cảm biến nhiệt độ:

Sử dụng cảm biến nhiệt độ LM35, loại cảm biến tương tự (analog) có khả năng đo nhiệt độ trong dải từ 0 đến 100°C với độ chính xác trung bình ±0.5°C. Đề tài không sử dụng cảm biến kỹ thuật số (DHT11, DS18B20), cảm biến hồng ngoại hoặc các module đo nhiệt độ không tiếp xúc.

- Chuyển đổi tín hiệu ADC:

Do AT89S52 không có ADC tích hợp, đề tài sử dụng IC ADC0804 để chuyển đổi tín hiệu nhiệt độ tương tự từ LM35 sang tín hiệu số. IC có độ phân giải 8 bit và yêu cầu thời gian chuyển đổi tối thiểu, phù hợp với mục tiêu đơn giản hóa.

- Quạt DC:

Điều khiển quạt DC loại nhỏ (12V), phù hợp với mạch điều khiển công suất thấp. Quạt được điều khiển thông qua xung PWM từ vi điều khiển (Timer0) và tín hiệu được khuếch đại qua transistor TIP122. Không áp dụng cho các loại quạt công nghiệp, quạt điện xoay chiều (AC), hoặc động cơ bước.

- Linh kiện giao tiếp và hiển thị:

Sử dụng màn hình LCD 1602 giao tiếp song song để hiển thị nhiệt độ thời gian thực. Đề tài không tích hợp các dạng giao tiếp hiện đại như I2C, SPI, UART hoặc các loại màn hình đồ họa như OLED, TFT.

- Cảnh báo nhiệt độ:

Chức năng cảnh báo khi vượt ngưỡng nhiệt độ được thực hiện thông qua buzzer đơn giản phát tiếng kêu cảnh báo. Không sử dụng các phương thức cảnh báo nâng cao như hiển thị màu sắc, tin nhắn, hoặc điều khiển từ xa.

1.3.2. Giới hạn về phần mềm và thuật toán

- Ngôn ngữ lập trình:

Hệ thống được lập trình hoàn toàn bằng ngôn ngữ Assembly trên phần mềm Keil C.

- Phương pháp điều khiển quạt:

Áp dụng phương pháp điều khiển bằng so sánh ngưỡng nhiệt độ, phân chia mức nhiệt thành các cấp tốc độ quạt (tăng dần theo từng mức), chứ chưa áp dụng điều khiển tuyến tính hoặc PID (Proportional – Integral – Derivative) hay điều khiển mờ (Fuzzy Logic).

- **Hiển thị dữ liệu:**

Chỉ hiển thị nhiệt độ tức thời, không có lưu trữ dữ liệu, biểu đồ lịch sử hay phân tích dữ liệu nhiệt theo thời gian.

- **Giao tiếp người dùng:**

Hệ thống là tự động hoàn toàn, không có nút nhấn điều chỉnh, không có tùy chỉnh cấu hình bởi người dùng hoặc điều khiển qua các thiết bị thông minh (điện thoại, máy tính).

1.3.3. Giới hạn trong triển khai thực tế

- **Quy mô thử nghiệm:**

Mô hình được thiết kế và kiểm thử trên mô phỏng Proteus và mạch thực nghiệm breadboard, chưa triển khai thành sản phẩm thương mại hay ứng dụng thực tế lâu dài.

- **Độ ổn định hệ thống:**

Mức độ chính xác và độ bền của hệ thống còn phụ thuộc vào nhiệt độ môi trường, điện áp cấp và chất lượng linh kiện. Hệ thống chưa được hiệu chuẩn với các thiết bị đo chuyên dụng.

- **Chưa tích hợp các tính năng nâng cao như:**

- Điều khiển từ xa qua Bluetooth, Wi-Fi
- Ứng dụng IoT (Internet of Things)
- Hệ thống quản lý năng lượng thông minh
- Cấu hình tùy chọn qua phần mềm

1.3.4. Giới hạn về thời gian và chi phí

- **Đề tài được thực hiện hơn 3 tháng từ ngày 04/02/2025 đến ngày 29/05/2025 dưới sự hướng dẫn của Ts. Đoàn Duy (Phó Trưởng Khoa Khoa Kỹ thuật Máy tính Trường ĐH Công nghệ Thông Tin – ĐHQG Tp.HCM).**

- **Chi phí thấp:**

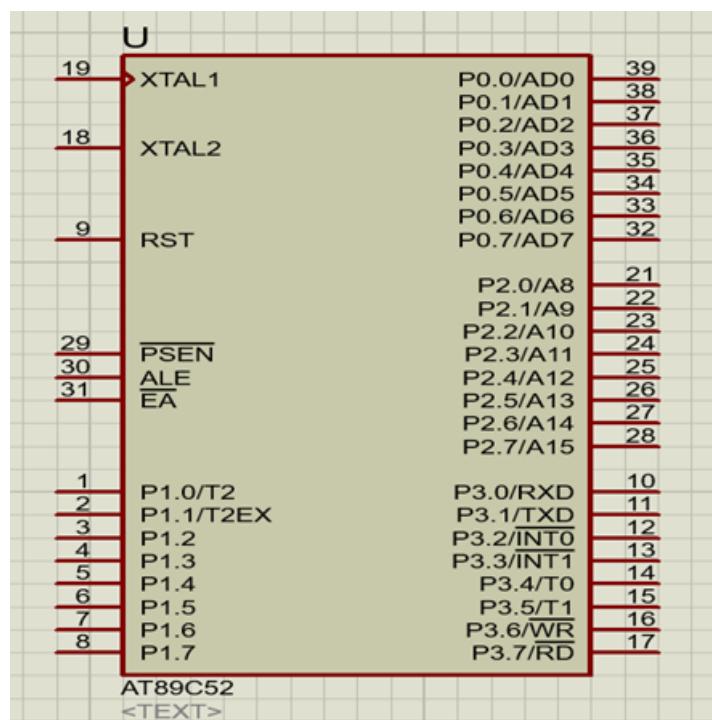
Chỉ sử dụng linh kiện phổ thông, giá rẻ, dễ mua, không sử dụng module phức tạp, vi điều khiển cao cấp hoặc phần cứng chuyên dụng.

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

2.1 CÁC LINH KIỆN SỬ DỤNG TRONG MẠCH

2.1.1 Vi điều khiển AT89S52

- Tổng quan



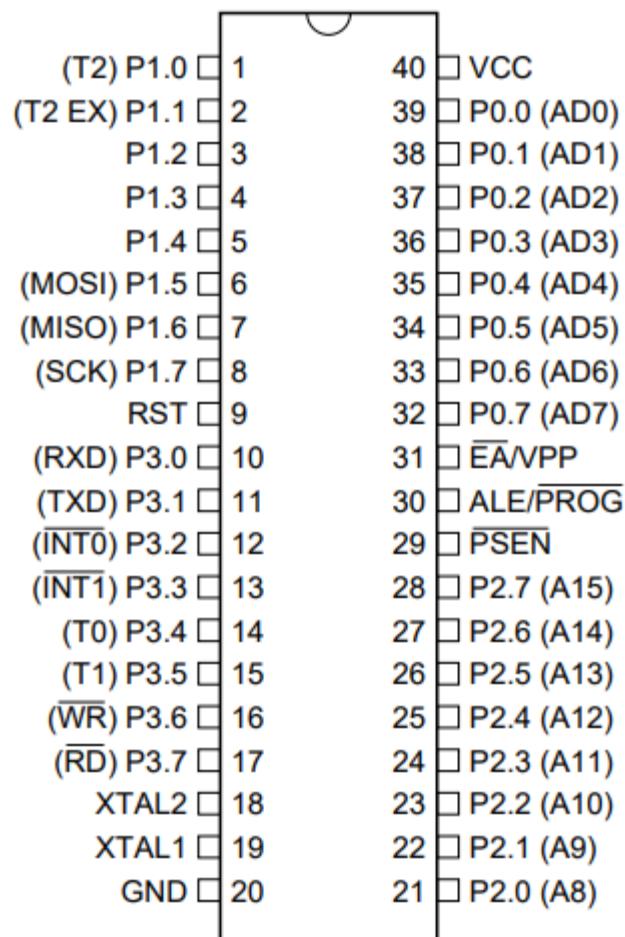
Hình 2.1.1 a Vi điều khiển AT89S52 với các chân

AT89S52 là một vi điều khiển (microcontroller) 8-bit thuộc họ 8051 do hãng Atmel (nay thuộc Microchip Technology) sản xuất. Đây là một phiên bản nâng cấp của vi điều khiển AT89C52, với một số tính năng nổi bật hơn.

AT89S52 cung cấp các tính năng tiêu chuẩn sau: 8K byte Flash, 256 byte RAM, 32 dòng I/O, bộ đếm thời gian Watchdog, hai con trỏ dữ liệu, ba bộ đếm thời gian/bộ đếm 16 bit, kiến trúc ngắt hai cấp sáu vectơ, cổng nối tiếp song công toàn phần, bộ dao động trên chip và mạch xung nhịp.

Ngoài ra, AT89S52 được thiết kế với logic tĩnh để hoạt động ở tần số bằng không và hỗ trợ hai chế độ tiết kiệm điện có thể lựa chọn bằng phần mềm.

- **I/O Port**



Hình 2.1.1b Vi điều khiển AT89S52 với các chân

- Sơ đồ chân VĐK AT89S52**

Bảng 2.1.1a Sơ đồ chân VĐK AT89S52

Chân	Tên chân	Chức năng chính	Chức năng phụ/mô tả
1 - 8	P1.0 - P1.7	Cổng I/O P1 (8 bit)	Giao tiếp với thiết bị. Ngoài ra, các chân P1.0 và P1.1 là hai chân liên quan đến hoạt động ngắt của bộ định thời 2.
9	RST	Reset	Đưa mức logic cao trong >2 chu kỳ máy để reset vi điều khiển.
10	P3.0 (RXD)	Cổng I/O	Nhận dữ liệu cho cổng nối tiếp
11	P3.1 (TXD)	Cổng I/O	Truyền dữ liệu cho cổng nối tiếp
12	P3.2 (INT0)	Cổng I/O	Ngắt ngoài 0
13	P3.3 (INT1)	Cổng I/O	Ngắt ngoài 1
14	P3.4 (T0)	Cổng I/O	Ngõ vào bộ đếm thời gian 0
15	P3.5 (T1)	Cổng I/O	Ngõ vào bộ đếm thời gian 1
16	P3.6 (WR)	Cổng I/O	Ghi dữ liệu đến bộ nhớ ngoài
17	P3.7 (RD)	Cổng I/O	Đọc dữ liệu đến bộ nhớ ngoài
18	XTAL2	Đao động thạch anh	Chân đầu ra của bộ tạo dao động bên ngoài

19	XTAL1	Dao động thạch anh	Chân đầu vào của bộ tạo dao động bên ngoài
20	GRD	Chân nối đất	Nối đất
21 - 28	P2.0 - P2.7	Cổng I/O	Phát địa chỉ cao từ A8 đến A15
29	PSEN	Program Store Enable	Dùng để nối chân OE của Rom ngoài
30	ALE/PROG	Address Latch Enable	Chốt địa chỉ byte thấp / Lập trình Flash.
31	EA/VPP	External Access	Mức thấp: dùng bộ nhớ chương trình ngoài Mức cao: dùng bộ nhớ trong.
32 - 39	P0.7 – P0.0	Cổng I/O	bus địa chỉ/dữ liệu khi dùng bộ nhớ ngoài (AD7– AD0)
40	VCC	Nguồn	Nguồn cấp điện 5V

Bảng 2.1.1b Mô tả chức năng các chân của AT89S52

- Các thanh ghi đặc biệt

Nhóm	Thanh ghi	Địa chỉ	Chức năng
CPU	ACC	-	Thanh ghi tích lũy, dùng cho phép toán số học và logic
	B	-	Thanh ghi phụ cho phép nhân và chia

	PSW	D0H	Thanh ghi trạng thái chương trình (CY, AC, F0, RS1, RS0, OV, P)
	SP	81H	Con trỏ ngăn xếp, chỉ định địa chỉ ngăn xếp trong RAM
	DPH, DPL	83H, 82H	Con trỏ dữ liệu 16-bit (DPTR) cho bộ nhớ dữ liệu hoặc chương trình ngoài
Interrupt	IE	A8H	Kích hoạt/vô hiệu hóa ngắt (EA, ET2, ES, ET1, EX1, ET0, EX0)
	IP	B8H	Xác định độ ưu tiên ngắt (cao/thấp)
Timer/ Counter	TCON	88H	Điều khiển Timer 0 và 1 (TF1, TR1, TF0, TR0, IE1, IT1, IE0, IT0)
	TMOD	89H	Cấu hình chế độ Timer 0 và 1 (GATE, C/T, M1, M0)
	TH0, TL0	8CH, 8AH	Thanh ghi đếm 8-bit cao/thấp của Timer 0
	TH1, TL1	8DH, 8BH	Thanh ghi đếm 8-bit cao/thấp của Timer 1
	T2CON	C8H	Điều khiển Timer 2 (TF2, EXF2, RCLK, TCLK, TR2, C/T2)
	TH2, TL2	CDH, CCH	Thanh ghi đếm 8-bit cao/thấp của Timer 2
	RCAP2H,	CBH,	Thanh ghi capture/reload cho Timer 2

	RCAP2L	CAH	
Serial (cổng nối tiếp)	SCON	98H	Điều khiển UART (SM0, SM1, SM2, REN, TB8, RB8, TI, RI)
	SBUF	99H	Bộ đệm dữ liệu nối tiếp (gửi/nhận dữ liệu)
I/O	P0	80H	Cổng vào/ra 8-bit, có thể dùng làm bus địa chỉ/dữ liệu
	P1	90H	Cổng vào/ra 8-bit, mục đích chung
	P2	A0H	Cổng vào/ra 8-bit, có thể dùng làm bus địa chỉ cao
	P3	B0H	Cổng vào/ra 8-bit, hỗ trợ chức năng đặc biệt (ngắt, timer, serial)

Bảng 2.1.1c Các thanh ghi đặc biệt của AT89S52

- **Tập lệnh**

Tập lệnh của vi điều khiển AT89S52 thuộc họ 8051 bao gồm khoảng 111 lệnh, được chia thành các nhóm chính: truyền dữ liệu, số học, logic, điều khiển chương trình, thao tác bit và lệnh đặc biệt. Các lệnh này hỗ trợ xử lý dữ liệu, tính toán, điều khiển I/O và quản lý chương trình hiệu quả, với độ dài lệnh từ 1 đến 3 byte và thời gian thực thi thường 1-2 chu kỳ máy. Tập lệnh được tối ưu cho ứng dụng nhúng, đặc biệt mạnh trong thao tác bit, phù hợp với các tác vụ như tạo thời gian trễ, giao tiếp UART, hoặc điều khiển thiết bị ngoại vi.

Nhóm lệnh	Chức năng	Ví dụ	Ứng dụng
Truyền dữ liệu	Di chuyển dữ liệu giữa thanh ghi, bộ nhớ trong/ngoài, cổng I/O.	MOV, MOVX, PUSH, POP, ...	Sao chép dữ liệu, truyền tham số, lưu trữ tạm thời trên ngăn xếp.

Số học	Thực hiện phép cộng, trừ, nhân, chia, tăng, giảm giá trị.	ADD, SUBB, MUL, INC, DEC, ...	Tính toán giá trị, điều chỉnh biến đếm, xử lý số học.
Logic	Thực hiện các phép toán logic (AND, OR, XOR, NOT), xoay bit.	ANL, CLR, RR, RRC, ...	Xử lý bit, tạo mặt nạ dữ liệu, mã hoá/giải mã.
Điều khiển chương trình	Điều khiển luồng chương trình: nhảy, gọi hàm, trả về.	NOP, RET, RETI, SJMP \$, ...	Vòng lặp, điều kiện, gọi hàm con, xử lý ngắt.
Thao tác bit	Thao tác trên các bit riêng lẻ trong bộ nhớ hoặc thanh ghi.	SETB, CLR, ...	Điều khiển I/O, kiểm tra trạng thái cờ, xử lý ngắt.
Đặc biệt	Các lệnh hệ thống như NOP, điều chỉnh BCD, hoán đổi bit.	NOP, DA, SWAP, ...	Tạo trễ ngắn, xử lý số BCD, hoán đổi dữ liệu.

Bảng 2.1.d Các nhóm lệnh của AT89S52

- **Ngắt**

Hệ thống ngắt của AT89S52 cho phép chương trình chính tạm dừng để thực thi một chương trình con ngắt (ISR - Interrupt Service Routine) khi một sự kiện cụ thể xảy ra. Sau khi xử lý ngắt, vi điều khiển quay lại thực thi chương trình chính từ điểm bị gián đoạn.

STT	Tên ngắt	Địa chỉ	Mô tả
1	INT0	0003H	Ngắt ngoài 0 (P3.2)
2	Timer 0	000BH	Ngắt tràn Timer 0
3	INT1	0013H	Ngắt ngoài 1 (P3.3)

4	TIMER 1	001BH	Ngắt tràn Timer 1
5	Serial	0023H	Ngắt truyền/nhận UART
6	Timer 2 (*)	002BH	Ngắt Timer 2 (trong AT89S52 - mở rộng)

Bảng 2.1.1e Cấu trúc hệ thống ngắt của AT89S52

- **THANH GHI QUẢN LÝ NGẮT**

Bảng 2.1.1f Bảng Thanh ghi quản lý ngắt

1. IE – Interrupt Enable (Địa chỉ: A8h)

Bit	Tên	Chức năng
EA	IE.7	Cho phép toàn bộ ngắt (1:bật)
-	IE.6	Chưa dùng
ET2	IE.5	Cho phép ngắt Timer 2
ES	IE.4	Cho phép ngắt Serial
ET1	IE.3	Cho phép ngắt Timer 1
EX1	IE.2	Cho phép ngắt ngoài INT1
ET0	IE.1	Cho phép ngắt Timer 0
EX0	IE.0	Cho phép ngắt ngoài INT0

2. IP – Interrupt Priority (Địa chỉ: B8h)

Cho phép xác định mức ưu tiên ngắt. Bit = 1: ưu tiên cao.

Bit	Tên	Ý nghĩa
-	IP.7-6	Chưa dùng
PT2	IP.5	Ưu tiên ngắt Timer 2
PS	IP.4	Ưu tiên ngắt Serial
PT1	IP.3	Ưu tiên ngắt Timer 1
PX1	IP.2	Ưu tiên ngắt ngoài INT1
PT0	IP.1	Ưu tiên ngắt Timer 0
PX0	IP.0	Ưu tiên ngắt ngoài INT0

Cơ chế hoạt động của ngắt:

1. Khi một sự kiện ngắt xảy ra, CPU kiểm tra xem:
 - o Cờ ngắt (flag) có được set chưa
 - o Bit cho phép ngắt có được bật không (trong IE)
2. Nếu có, chương trình tự động nhảy đến địa chỉ vector ngắt tương ứng.
3. Chương trình phục vụ ngắt (ISR) được thực hiện.
4. Sau khi kết thúc, lệnh RETI sẽ đưa chương trình quay lại vị trí trước khi ngắt xảy ra.

- **Lập trình Vi điều khiển AT89S52**

Vi điều khiển AT89S52 có thể được lập trình theo hai phương thức: lập trình song song (Parallel Programming) và lập trình nối tiếp (Serial Programming). Trong đó, chế độ lập trình nối tiếp – còn gọi là ISP (In-System Programming) – là phương pháp phổ biến nhất nhờ ưu điểm dễ thực hiện, tiện lợi và không cần thiết bị chuyên dụng phức tạp. Việc lập trình nối tiếp

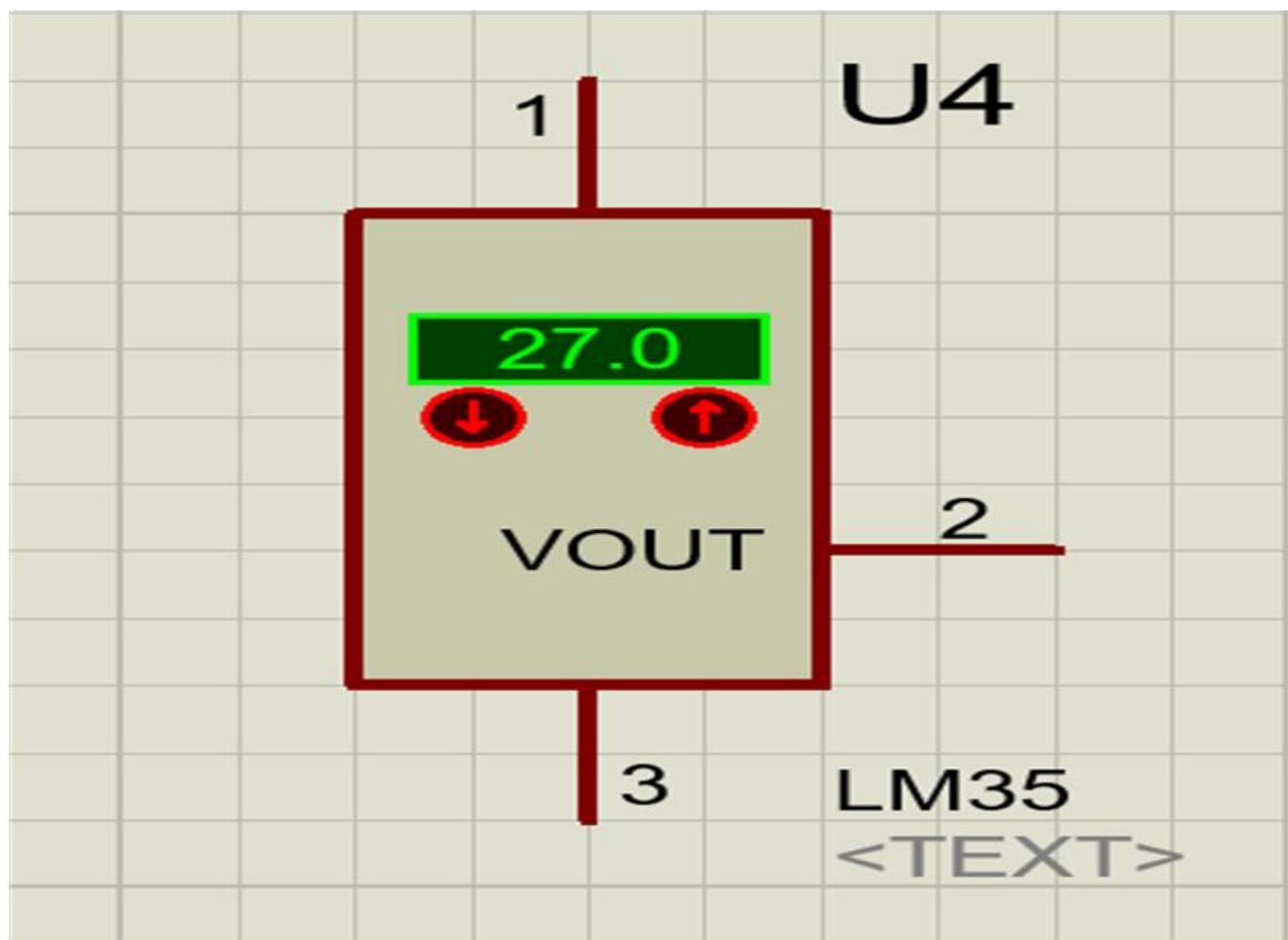
sử dụng giao thức SPI, cho phép nạp chương trình trực tiếp vào bộ nhớ Flash thông qua các chân giao tiếp chuyên dụng.

Trong các ứng dụng thực tế, mạch nạp ISP như USBasp thường được sử dụng để lập trình AT89S52. Đồ án này sử dụng chính phương pháp lập trình nối tiếp để nạp chương trình vào vi điều khiển. Các bước thực hiện lập trình qua giao tiếp SPI được mô tả như sau:

1. Cấp nguồn cho vi điều khiển:
 - Cấp điện áp ổn định giữa chân VCC và GND.
 - Đưa chân RST lên mức cao (logic 1) để kích hoạt chế độ lập trình.
2. Bắt đầu chế độ lập trình nối tiếp:
 - Gửi lệnh kích hoạt lập trình (Programming Enable) vào chân MOSI (P1.5).
 - Đồng thời, cấp xung nhịp vào chân SCK (P1.7) với tần số nhỏ hơn xung hệ thống – thường là xung chia 16 so với xung nhịp CPU.
3. Ghi dữ liệu chương trình vào Flash:
 - Mã máy (code) được ghi vào bộ nhớ Flash theo từng byte hoặc theo từng trang (page mode).
 - Mỗi chu kỳ ghi thường cần giữ mức áp thích hợp khoảng 0.5 ms ở mức 5V để đảm bảo dữ liệu ghi ổn định.
4. Kiểm tra dữ liệu đã ghi (Verify):
 - Có thể xác minh bất kỳ địa chỉ nào trong bộ nhớ bằng cách gửi lệnh đọc nội dung.
 - Dữ liệu từ bộ nhớ sẽ được truyền về qua chân MISO (P1.6) để kiểm tra tính toàn vẹn.
5. Kết thúc quá trình lập trình:
 - Sau khi hoàn tất quá trình ghi và kiểm tra, đưa chân RST về mức thấp để vi điều khiển trở lại chế độ hoạt động bình thường.

2.1.2 Cảm biến nhiệt độ LM35

Tổng quan



Hình 2.1.2 Cảm biến LM35

Cảm biến nhiệt độ LM35 là một cảm biến analog được thiết kế để đo nhiệt độ môi trường với độ chính xác cao, thường được sử dụng trong các ứng dụng nhúng và điều khiển tự động. Được sản xuất bởi Texas Instruments, không giống như các cảm biến nhiệt độ dạng điện trở (như thermistor hay RTD), LM35 cho tín hiệu điện áp tuyến tính tỉ lệ trực tiếp với nhiệt độ theo đơn vị $^{\circ}\text{C}$, thuận tiện cho việc xử lý trong vi điều khiển thông qua mạch chuyển đổi ADC.

Thông số kỹ thuật

- Phạm vi nhiệt độ: -55°C đến $+150^{\circ}\text{C}$.
- Độ phân giải: Đầu ra $10 \text{ mV}/^{\circ}\text{C}$ (ví dụ: 25°C tương ứng với 250 mV).
- Điện áp nguồn: 4V đến 30V (thường dùng 5V trong mạch vi điều khiển).
- Độ chính xác:
 - $\pm 0.5^{\circ}\text{C}$ tại 25°C (phiên bản tiêu chuẩn).
 - $\pm 1^{\circ}\text{C}$ trong phạm vi -10°C đến $+85^{\circ}\text{C}$.
- Dòng tiêu thụ: Khoảng $60 \mu\text{A}$.
- Gói chân: TO-92 (3 chân: Vcc, Vout, GND).
- Ưu điểm: Không cần hiệu chỉnh, phản ứng nhanh, dễ tích hợp với vi điều khiển như AT89S52.

Cấu tạo chân và nguyên lý hoạt động

- Cấu tạo chân:

Số chân	Tên chân	Chức năng
1	V _{CC} hay +V _S	Chân cấp nguồn với điện áp từ 4V đến 30V
2	V _{OUT}	Chân lấy điện áp ra, điện áp ở chân này thay đổi $10\text{mV}/^{\circ}\text{C}$
3	GND	Chân nối đất

Bảng 2.1.2 Cấu tạo cảm biến LM35

- Nguyên lý hoạt động:

LM35 hoạt động dựa trên hiệu ứng nhiệt điện, trong đó điện áp đầu ra thay đổi tuyến tính với nhiệt độ môi trường. Cụ thể:

- Đầu ra (V_{out}) tăng 10 mV cho mỗi độ Celsius tăng lên.
- Công thức tính nhiệt độ: [$T ({}^\circ C) = \frac{V_{out}}{10} \times 1000 / 10 = V_{out} / 10$] Ví dụ: Nếu $V_{out} = 300$ mV, thì ($T = 300 / 10 = 30 {}^\circ C$).
- Cảm biến sử dụng một mạch tích hợp để đo sự thay đổi nhiệt độ và chuyển đổi thành tín hiệu analog, phù hợp để xử lý bởi ADC (Analog-to-Digital Converter) của vi điều khiển.

Kết nối LM35 với vi điều khiển

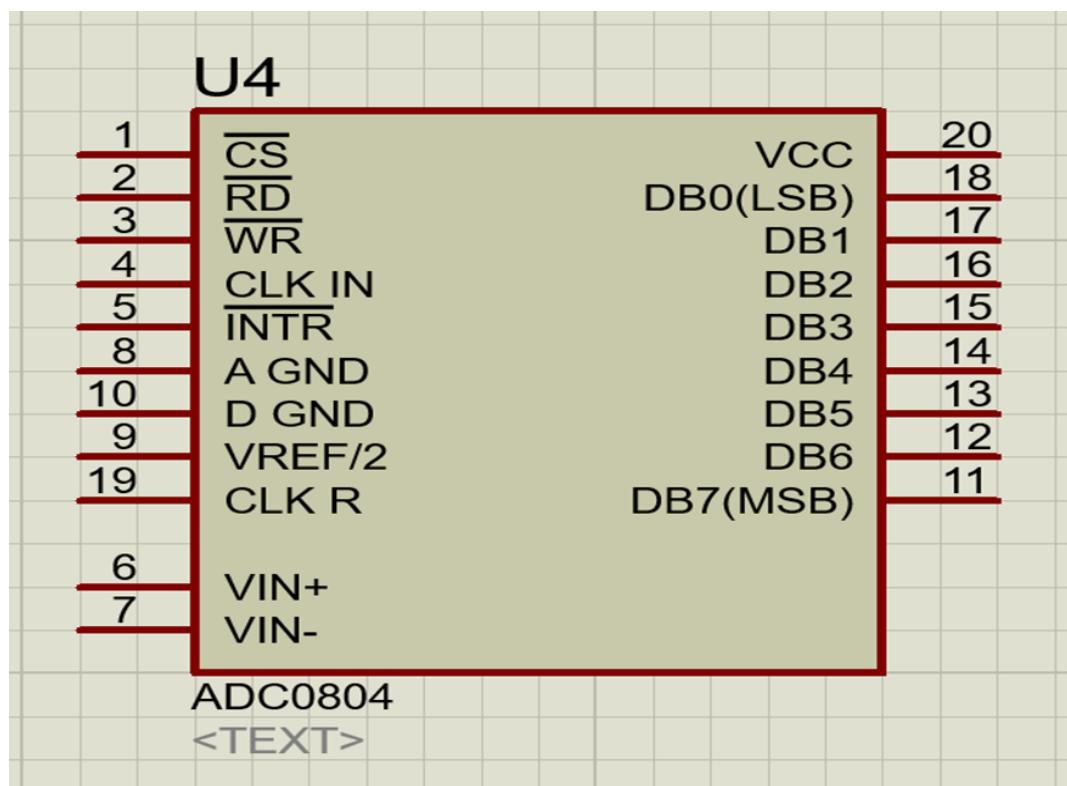
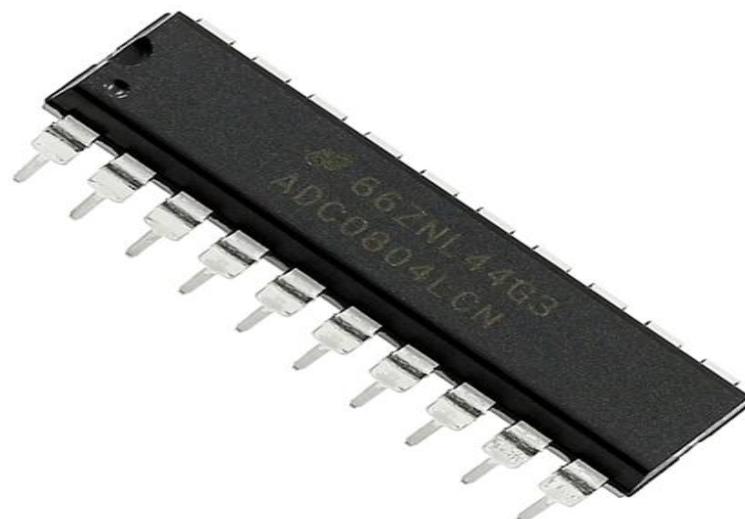
- Trường hợp vi điều khiển có ADC tích hợp:

Ví dụ: PIC, AVR, STM32 → LM35 có thể được nối trực tiếp vào chân ADC.

- Trường hợp vi điều khiển KHÔNG có ADC (như AT89S52):
- Cần sử dụng thêm IC chuyển đổi ADC rời, như ADC0804
- LM35 sẽ cấp tín hiệu analog cho ADC0804 → vi điều khiển đọc dữ liệu digital từ ADC qua cổng I/O (P0, P1...)

2.1.3 Bộ chuyển đổi ADC0804

Tổng quan



Hình 2.1.3 Bộ chuyển đổi ADC0804

Bộ chuyển đổi **ADC0804** là một bộ chuyển đổi Analog-to-Digital (ADC) 8-bit phổ biến, được sản xuất bởi Texas Instruments, thường được sử dụng trong các hệ thống nhúng để chuyển đổi tín hiệu analog thành tín hiệu số. ADC0804 được thiết kế để hoạt động với vi điều khiển như AT89S52, cung cấp khả năng đọc tín hiệu từ cảm biến analog (như LM35) một cách hiệu quả.

Thông số kỹ thuật

Thông số	Giá trị
Độ phân giải	8 bit (256 mức từ 0 -255)
Điện áp hoạt động	5V ±10%
Dải điện áp đầu vào	0 – 5V
Tốc độ chuyển đổi	~100 µs
Số kênh đầu vào	1 kênh analog
Giao tiếp dữ liệu	Dạng song song 8 bit
Đóng gói	DIP 20 chân

Bảng 2.1.3a Thông số kỹ thuật ADC804

Cấu tạo chân và nguyên lý hoạt động

- Cấu tạo chân:

Chân	Tên	Chức năng chính
1	CS	Chọn chip (chân kích hoạt)
2	RD	Đọc dữ liệu từ ADC (khi ở mức thấp)
3	WR	Gửi lệnh bắt đầu chuyển đổi ADC

4	CLK IN	Tín hiệu xung clock vào
5	IN (+)	Ngõ vào tín hiệu analog (chân dương)
6	IN (-)	Ngõ vào âm (thường nối GND)
7	VREF/2	Điều chỉnh độ phân giải (thường nối 2.5V hoặc bỏ trống)
8	GND	Nối đất
9–16	D0–D7	Dữ liệu ra dạng song song (8 bit)
17	INT	Chân ngắt báo chuyển đổi hoàn tất
18	CLK R	Clock nội (dùng RC bên ngoài)
19	VCC	Nguồn cấp (5V)

Bảng 2.1.3b Thông số kỹ thuật ADC804

- Nguyên lý hoạt động:

Quy trình chuyển đổi gồm các bước chính:

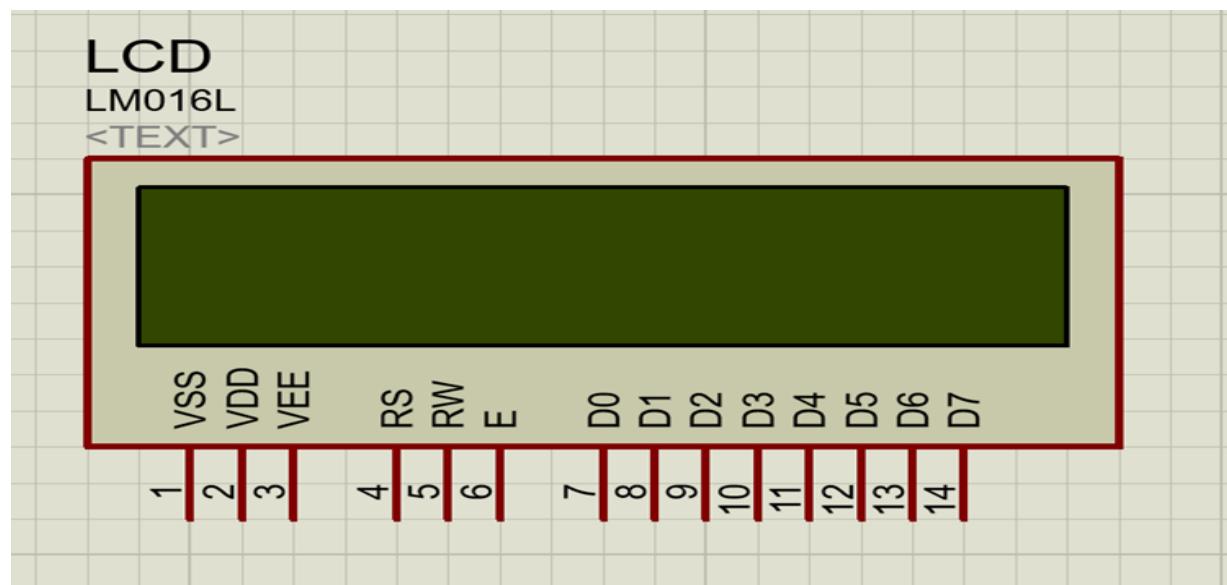
1. Kích hoạt bắt đầu chuyển đổi:
 - Đưa chân WR xuống mức thấp (0) trong một thời gian ngắn → ADC bắt đầu chuyển đổi tín hiệu analog.
 - Sau đó đưa WR lên lại mức cao.
2. Đợi hoàn thành chuyển đổi:
 - ADC0804 tự động xử lý tín hiệu analog.
 - Sau khi xong, chân INT sẽ chuyển sang mức thấp (báo hiệu hoàn tất).
3. Đọc dữ liệu:
 - Đưa chân RD xuống mức thấp → dữ liệu digital (8 bit) sẽ xuất hiện tại các chân D0 đến D7.
 - Đọc giá trị này bằng vi điều khiển.

Kết nối ADC0804 với Vi điều khiển

- Chân IN+ nối với đầu ra cảm biến (VD: LM35)
- D0–D7 nối với một port của vi điều khiển (P0 hoặc P1)
- CS, RD, WR nối với các chân điều khiển (có thể dùng P2)
- INT nối đến chân ngắt hoặc cổng kiểm tra trạng thái
- CLK IN có thể tạo bằng thạch anh hoặc RC đơn giản

2.1.4 Màn hình LCD 16x02

Tổng quan



Hình 2.1.4 Màn hình LCD 16x2

Màn hình LCD 16x2 (Liquid Crystal Display) là một loại màn hình tinh thể lỏng phổ biến, thường được sử dụng trong các ứng dụng nhúng để hiển thị thông tin dạng văn bản. Với cấu trúc 16 ký tự trên 2 dòng (tổng cộng 32 ký tự), LCD 16x2 được tích hợp với vi điều khiển như AT89S52 để hiển thị dữ liệu như nhiệt độ, thời gian, hoặc thông báo trạng thái.

Thông số kỹ thuật

- Kích thước hiển thị: 16 ký tự x 2 dòng (16x2).
- Loại ký tự: 5x8 điểm ảnh (mỗi ký tự là ma trận 5x8 pixel).
- Bộ điều khiển tích hợp: Thường sử dụng chip HD44780 hoặc tương thích (như ST7066U).
- Điện áp hoạt động: 4.7V đến 5.3V (thường dùng 5V).
- Dòng tiêu thụ:
 - Khoảng 1 mA (không tính đèn nền).
 - Đèn nền LED tiêu thụ thêm 10-20 mA (tùy chỉnh qua biến trờ).
- Giao tiếp: Ché độ 4-bit hoặc 8-bit (thường dùng 4-bit để tiết kiệm chân I/O).
- Kích thước vật lý: Khoảng 80mm x 36mm x 10mm (tùy model).
- Nhiệt độ hoạt động: 0°C đến 50°C (phiên bản tiêu chuẩn).
- Đèn nền: Có (thường là LED xanh hoặc trắng), điều chỉnh độ sáng bằng biến trờ.
- Ưu điểm: Giá rẻ, dễ lập trình, hiển thị rõ ràng, tiêu thụ điện thấp.

Cấu tạo chân và nguyên lý hoạt động

- Cấu tạo chân và chức năng:

LCD 16x2 có 16 chân, với chức năng cụ thể như sau:

Chân	Tên	Chức năng
1	VSS (GND)	Chân nối GND
2	VDD (VCC)	Nguồn dương (+5V)

3	V0	Điều chỉnh độ tương phản (nối qua biến trở $10k\Omega$, thường từ 0V đến 5V)
4	RS	Chọn thanh ghi (0: lệnh, 1: dữ liệu)
5	RW	Đọc/Ghi (0: ghi, 1: đọc, thường nối GND vì chỉ ghi dữ liệu).
6	E	Enable: Chân kích xung cho LCD nhận lệnh/dữ liệu
7 - 14	D0 - D7	8 chân dữ liệu (có thể chỉ dùng D4-D7 nếu dùng chế độ 4-bit)
15	LED+	Cực dương của đèn nền (nối với 5V qua điện trở 220Ω để giới hạn dòng).
16	LED-	Cực âm của đèn nền (nối với GND).

Bảng 2.1.4a Cấu tạo chân LCD 16x2

- **Nguyên lý hoạt động:**

LCD 16x2 hoạt động dựa trên công nghệ tinh thể lỏng, sử dụng bộ điều khiển HD44780 để xử lý các lệnh và dữ liệu từ vi điều khiển:

1. Tinh thể lỏng: Các phân tử tinh thể lỏng được điều khiển bởi điện áp để cho phép hoặc chặn ánh sáng đi qua, tạo thành các ký tự.
2. Bộ điều khiển HD44780:
 - Nhận lệnh từ vi điều khiển (ví dụ: xóa màn hình, di chuyển con trỏ).
 - Nhận dữ liệu để hiển thị (ký tự ASCII).
 - Điều khiển ma trận điểm ảnh 5x8 để hiển thị từng ký tự.
3. Giao tiếp:
 - Chế độ 8-bit: Sử dụng cả 8 chân dữ liệu (DB0-DB7) để truyền dữ liệu/lệnh.
 - Chế độ 4-bit: Truyền dữ liệu/lệnh bằng cách gửi 4 bit cao trước, sau đó 4 bit thấp, qua DB4-DB7.
4. Đèn nền: LED nền tạo ánh sáng để ký tự dễ nhìn, độ sáng được điều chỉnh qua chân LED+ và LED-.

Quy trình hiển thị:

- Khởi tạo LCD (cài đặt chế độ, xóa màn hình, bật hiển thị).
- Gửi lệnh (ví dụ: đặt con trỏ).
- Gửi dữ liệu (ký tự cần hiển thị).

Bảng 2.1.4b Các lệnh cơ bản của LCD 16x02

Lệnh	Mã lệnh (Hex)	Chức năng
Clear display	0x01	Xóa toàn bộ màn hình
Return home	0x02	Đưa con trỏ về đầu dòng
Entry mode set	0x06	Tự động dịch con trỏ sau khi in ký tự
Display ON/OFF	0x0C	Hiển thị màn hình, tắt con trỏ
Cursor ON/OFF	0x0E	Hiển thị con trỏ nhấp nháy
Function set	0x38 hoặc 0x28	Thiết lập chế độ 8-bit (0x38) hoặc 4-bit (0x28)
Set DDRAM address	0x80 + addr	Đặt vị trí con trỏ (ví dụ: 0x80 đầu dòng 1)

Kết nối LCD 16x02 với Vi điều khiển

LCD 16x2 thường được kết nối với AT89S52 ở chế độ 4-bit để tiết kiệm chân I/O. Dưới đây là cách đấu nối:

- Chân LCD:
 - VSS (chân 1): Nối GND.
 - VDD (chân 2): Nối 5V.
 - V0 (chân 3): Nối với biến trở $10k\Omega$ (để điều chỉnh tương phản).

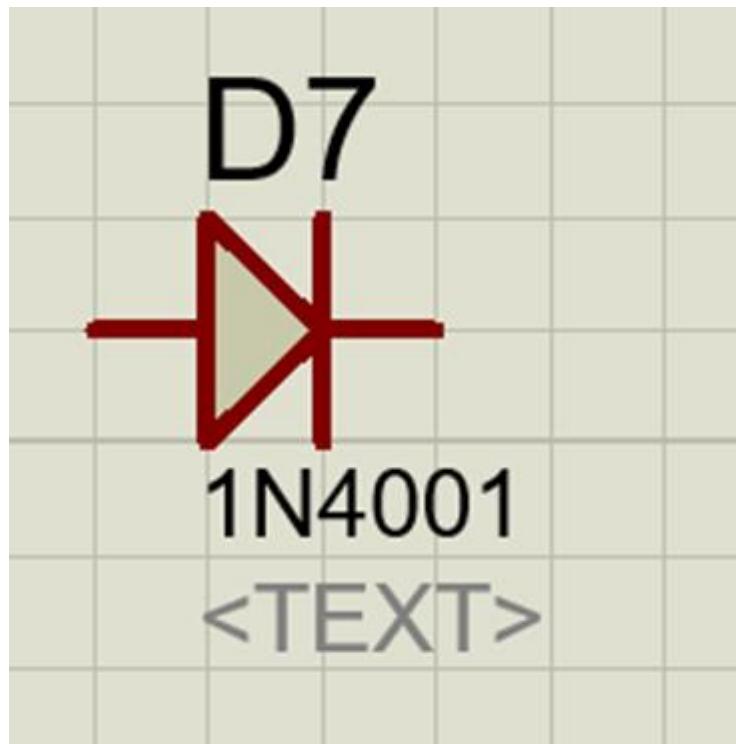
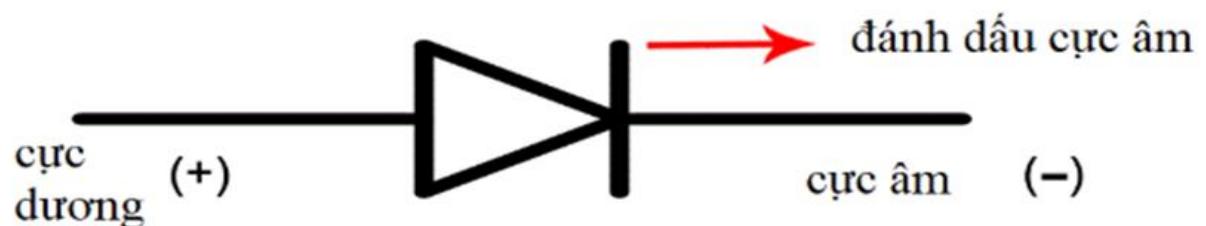
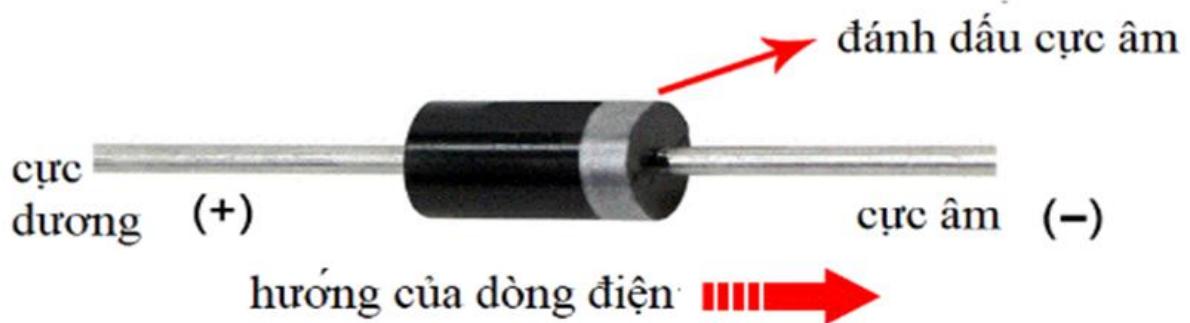
- RS (chân 4): Nối P2.0.
- RW (chân 5): Nối GND (chỉ ghi dữ liệu).
- E (chân 6): Nối P2.1.
- DB4-DB7 (chân 11-14): Nối P1.0-P1.3.
- DB0-DB3 (chân 7-10): Không sử dụng (chế độ 4-bit).
- LED+ (chân 15): Nối 5V qua điện trở 220Ω .
- LED- (chân 16): Nối GND.
- Nguồn và mass: Đảm bảo chung nguồn với AT89S52.

Sơ đồ đầu nối:

- P2.0 (RS), P2.1 (E) điều khiển tín hiệu.
- P1.0-P1.3 truyền dữ liệu 4-bit.
- Biến trở $10k\Omega$ điều chỉnh độ tương phản qua V0.
- Đèn nền LED được cấp nguồn 5V qua điện trở 220Ω .

2.1.5 Diode 1N4001

Tổng quan



Hình 2.1.5 Diode 1N4001

Diode 1N4001 là một trong những loại diode chỉnh lưu phổ biến nhất trong dòng 1N400x, được sử dụng rộng rãi trong các mạch điện tử nhờ vào độ tin cậy, giá thành thấp và khả năng hoạt động hiệu quả trong các ứng dụng chỉnh lưu dòng điện. Được phát triển bởi các nhà sản xuất linh kiện điện tử hàng đầu như ON Semiconductor, Vishay, và Diodes Incorporated, 1N4001 là lựa chọn lý tưởng cho các ứng dụng điện áp thấp và dòng điện trung bình.

Đặc điểm kỹ thuật của Diode 1N4001

1. Thông số kỹ thuật chính

Diode 1N4001 là diode chỉnh lưu silicon thuộc dòng 1N400x, với các thông số kỹ thuật chính như sau:

- Điện áp ngược định mức tối đa (VRRM): 50V
- Dòng điện chỉnh lưu trung bình tối đa (IO): 1A
- Dòng điện xung không lặp lại (IFSM): 30A (trong 8.3ms, sóng hình sin)
- Điện áp thuận tối đa (VF): 1.1V (ở dòng điện 1A)
- Dòng điện ngược tối đa (IR): 5 μ A (ở 50V, 25°C)
- Nhiệt độ hoạt động: -55°C đến +150°C
- Kích thước và vỏ bọc: DO-41 (DO-204AL), vỏ nhựa cách điện
- Điện dung tiếp giáp (Cj): Khoảng 15pF (ở tần số 1MHz)

2. Cấu tạo vật liệu

Diode 1N4001 được chế tạo từ vật liệu bán dẫn silicon, với cấu trúc tiếp giáp PN tiêu chuẩn. Vỏ bọc DO-41 bằng nhựa epoxy cung cấp khả năng cách điện tốt và bảo vệ diode khỏi các yếu tố môi trường như độ ẩm và bụi. Hai chân dẫn (anode và cathode) được làm từ hợp kim chống ăn mòn, đảm bảo khả năng hàn tốt và độ bền cơ học.

3. Đặc tính điện

Đặc tính chỉnh lưu: Diode 1N4001 chỉ cho phép dòng điện chạy theo một chiều từ anode đến cathode, ngăn chặn dòng điện ngược, làm cho nó phù hợp với các ứng dụng chỉnh lưu.

Đáp ứng tần số: Phù hợp cho các ứng dụng tần số thấp như chỉnh lưu nguồn 50/60Hz.

Khả năng chịu nhiệt: Với dải nhiệt độ hoạt động rộng, diode có thể hoạt động ổn định trong các môi trường khắc nghiệt.

Bảng 2.1.5 So sánh với các diode khác cùng dòng

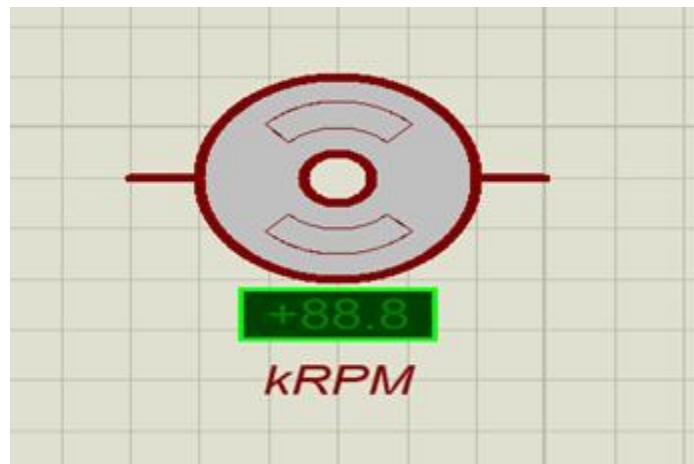
Loại diode	Điện áp ngược cực đại
1N4001	50V
1N4002	100V
1N4004	400V
1N4007	1000V

Nguyên lý hoạt động

- Khi phân cực thuận (Anode dương, Cathode âm): Diode dẫn điện, dòng điện đi qua diode.
- Khi phân cực ngược (Anode âm, Cathode dương): Diode không dẫn điện (chỉ có dòng rò nhỏ), giúp ngăn dòng điện ngược gây hư hại cho mạch.

2.1.6 Fan - DC

Tổng quan





Hình 2.1.6 Fan DC

Quạt DC (Direct Current Fan) là loại quạt sử dụng động cơ chạy bằng dòng điện một chiều (DC), khác với quạt AC (Alternating Current) sử dụng dòng điện xoay chiều. Với sự phát triển của công nghệ, quạt DC ngày càng được ưa chuộng nhờ vào hiệu suất năng lượng cao, khả năng hoạt động êm ái và tính linh hoạt trong điều chỉnh tốc độ. Quạt DC được sử dụng phổ biến trong các ứng dụng gia dụng, công nghiệp và điện tử, từ quạt làm mát không gian đến quạt trong các thiết bị điện tử như máy tính.

Cấu tạo

Quạt DC có cấu tạo tương tự các loại quạt thông thường nhưng được thiết kế để hoạt động với động cơ DC. Các thành phần chính bao gồm:

1. Động cơ DC:

- Là bộ phận cốt lõi, thường sử dụng động cơ không chổi than (Brushless DC Motor - BLDC) để tăng hiệu suất và độ bền.
- Động cơ bao gồm stator (phản tĩnh với cuộn dây) và rotor (phản quay với nam châm vĩnh cửu).
- Một mạch điều khiển điện tử (driver) được tích hợp để điều chỉnh dòng điện và tốc độ quay của động cơ.

2. Cánh quạt:

- Được làm từ nhựa, kim loại hoặc composite, thiết kế tối ưu để tạo luồng gió mạnh và ổn định.
- Số lượng cánh (thường 3-5 cánh) và hình dạng cánh ảnh hưởng đến lưu lượng gió và độ ồn.

3. Lồng quạt:

- Bảo vệ cánh quạt, ngăn chặn vật thể lạ xâm nhập và đảm bảo an toàn cho người sử dụng.
- Thường làm từ kim loại hoặc nhựa bền.

4. Thân và đế quạt:

- Cung cấp cấu trúc cố định cho quạt, có thể là quạt bàn, quạt đứng, quạt treo tường hoặc quạt trần.
- Đế quạt thường được thiết kế chắc chắn để đảm bảo độ ổn định.

5. Bộ nguồn và mạch điều khiển:

- Quạt DC yêu cầu nguồn điện một chiều (thường 12V, 24V hoặc 48V), có thể được cung cấp từ bộ nguồn AC-DC, pin hoặc ắc quy.
- Mạch điều khiển cho phép điều chỉnh tốc độ quay, hẹn giờ và các tính năng hiện đại như điều khiển từ xa.

6. Các bộ phận phụ:

- Ốc vít, vòng bi (bearing) để giảm ma sát, và các cảm biến (như cảm biến nhiệt độ) trong một số mẫu quạt cao cấp.

Nguyên lý hoạt động

Quạt DC hoạt động dựa trên nguyên lý chuyển đổi năng lượng điện thành năng lượng cơ học thông qua động cơ DC. Cụ thể:

1. Động cơ không chổi than (BLDC):

- Dòng điện một chiều được cung cấp cho cuộn dây trên stator thông qua mạch điều khiển điện tử.
- Từ trường sinh ra từ cuộn dây tương tác với nam châm vĩnh cửu trên rotor, tạo ra lực quay.
- Mạch điều khiển sử dụng kỹ thuật PWM (Pulse Width Modulation) để điều chỉnh tốc độ quay bằng cách thay đổi chu kỳ làm việc của dòng điện.

2. Tạo luồng gió:

- Rotor quay làm quay các cánh quạt, đẩy không khí di chuyển để tạo ra luồng gió.
- Hướng gió thường song song với trục quay (quạt hướng trục) hoặc vuông góc (quạt ly tâm), tùy thuộc vào thiết kế.

3. Chế độ AC/DC (trong một số mẫu quạt tích điện):

- Một số quạt DC tích hợp chế độ AC/DC, cho phép chuyển đổi linh hoạt giữa nguồn điện xoay chiều (AC) và một chiều (DC) thông qua bộ chuyển đổi nguồn hoặc ắc quy tích hợp. Chế độ DC đảm bảo hoạt động êm ái và tiết kiệm năng lượng, trong khi chế độ AC cung cấp nguồn điện liên tục khi cần.

Bảng 2.1.6a So sánh quạt DC và quạt AC

Tiêu chí	Quạt DC	Quạt AC
Nguồn điện	Dòng điện một chiều (DC)	Dòng điện xoay chiều (AC)
Hiệu suất năng lượng	Cao, tiết kiệm 30-50% điện	Thấp hơn, tiêu thụ điện nhiều hơn
Độ ồn	Thấp (20-40 dB)	Cao hơn (30-60 dB)

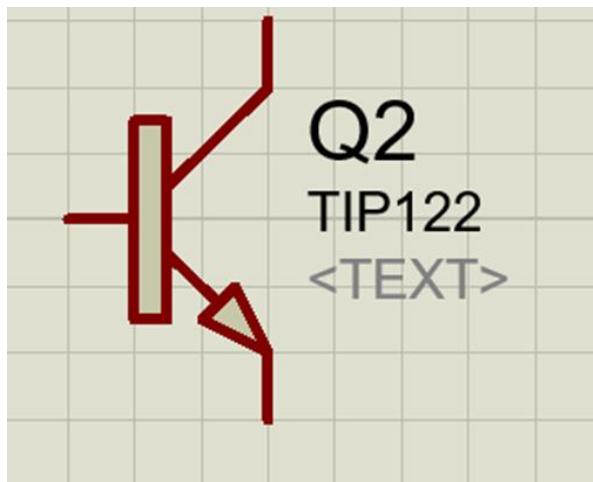
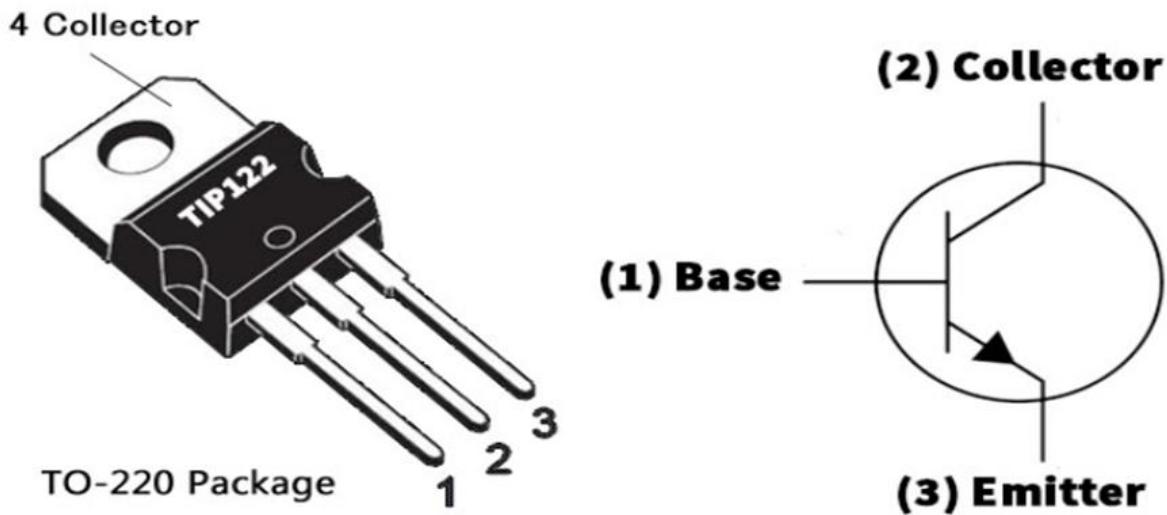
Điều chỉnh tốc độ	Linh hoạt, nhiều cấp độ qua PWM	Hạn chế, thường 3-5 cấp độ
Giá thành	Cao hơn	Thấp hơn
Tuổi thọ	Cao (50.000-100.000 giờ)	Thấp hơn (20.000-50.000 giờ)
Ứng dụng	Gia dụng, điện tử, năng lượng tái tạo	Gia dụng, công nghiệp cơ bản

Bảng 2.1.6b Phân loại Quạt DC

Loại quạt DC	Đặc điểm chính
Quạt có chổi than	Giá rẻ, cấu tạo đơn giản, tuổi thọ ngắn hơn
Quạt không chổi than	Bền hơn, êm hơn, hiệu suất cao, thường tích hợp IC điều tốc
Quạt có cảm biến Hall	Có tín hiệu phản hồi tốc độ (tachometer) – hỗ trợ giám sát RPM
Quạt PWM (4 dây)	Điều khiển tốc độ bằng xung PWM – phổ biến trong máy tính và thiết bị công nghiệp

2.1.7 Transistor TIP122

Tổng quan



Hình 2.1.7 Transistor TIP122

Transistor TIP122 là một transistor lưỡng cực NPN (Bipolar Junction Transistor - BJT) thuộc dòng Darlington, được thiết kế để xử lý dòng điện và công suất cao. Đây là một linh kiện bán dẫn phô biến, được sản xuất bởi các hãng như ON Semiconductor, STMicroelectronics và Texas Instruments. Nhờ cấu hình Darlington, TIP122 cung cấp hệ số khuếch đại dòng điện cao, phù hợp cho các ứng dụng yêu cầu điều khiển tải lớn, chẳng hạn như trong mạch điều khiển động cơ, bộ nguồn hoặc hệ thống công suất.

Thông số kỹ thuật

- Cấu tạo chân

Bảng 2.1.7 TIP122 có 3 chân

Chân	Ký hiệu	Chức năng
1	Base (B)	Cực điều khiển
2	Collector (C)	Cực thu (nối với tải)
3	Emitter (E)	Cực phát

Đặc điểm kỹ thuật của TIP122

1. Thông số kỹ thuật chính

Transistor TIP122 có các thông số kỹ thuật quan trọng như sau:

- Loại transistor: NPN Darlington
- Điện áp định mức tối đa giữa Collector và Emitter (V_{CEO}): 100V
- Điện áp định mức tối đa giữa Collector và Base (V_{CBO}): 100V
- Điện áp định mức tối đa giữa Emitter và Base (V_{EBO}): 5V
- Dòng điện Collector liên tục tối đa (I_C): 5A
- Dòng điện Collector xung tối đa (I_{CM}): 8A
- Công suất tiêu tán tối đa (P_D): 65W (với tản nhiệt phù hợp, ở $25^\circ C$)
- Hệ số khuếch đại dòng điện (h_{FE}): 1000 (tối thiểu) ở $I_C = 3A$, $V_{CE} = 3V$
- Điện áp bão hòa Collector-Emitter ($V_{CE(sat)}$): 2V (tối đa) ở $I_C = 3A$, $I_B = 12mA$
- Nhiệt độ hoạt động: $-65^\circ C$ đến $+150^\circ C$
- Vỏ bọc: TO-220 (gói công suất, có lỗ gắn tản nhiệt)
- Tần số chuyển mạch: Thấp, phù hợp với các ứng dụng tần số thấp (dưới 1MHz)

2. Cấu tạo vật liệu

- Vật liệu bán dẫn: Silicon, đảm bảo hiệu suất cao và độ bền nhiệt tốt.

- Cấu hình Darlington: TIP122 bao gồm hai transistor NPN được kết nối theo cấu hình Darlington, với một transistor điều khiển (driver) và một transistor công suất. Điều này làm tăng hệ số khuếch đại dòng điện (h_{FE}) nhưng cũng làm tăng điện áp bão hòa.
- Vỏ bọc: TO-220, làm từ nhựa epoxy với tám kim loại phía sau để gắn tản nhiệt, giúp tản nhiệt hiệu quả trong các ứng dụng công suất cao.
- Chân kết nối: Bao gồm 3 chân (Base, Collector, Emitter), trong đó chân Collector thường được nối với tám kim loại của vỏ TO-220.

3. Đặc tính điện

- Hệ số khuếch đại cao: Nhờ cấu hình Darlington, TIP122 có h_{FE} lớn (tối thiểu 1000), cho phép điều khiển dòng điện lớn với dòng Base nhỏ.
- Điện áp bão hòa thấp: $V_{CE(sat)}$ tối đa 2V, giúp giảm tổn hao năng lượng trong trạng thái dẫn.
- Khả năng chịu dòng và điện áp cao: Phù hợp cho các ứng dụng công suất trung bình như điều khiển động cơ hoặc tải cảm ứng.

Nguyên lý hoạt động

Transistor TIP122 hoạt động như một công tắc hoặc bộ khuếch đại dòng điện trong mạch điện tử:

1. Cấu hình Darlington:

- TIP122 bao gồm hai transistor NPN nối tiếp, trong đó dòng Base của transistor thứ nhất điều khiển transistor thứ hai. Điều này làm tăng đáng kể hệ số khuếch đại dòng điện ($h_{FE} = h_{FE1} \times h_{FE2}$).
- Dòng Base nhỏ (I_B) có thể điều khiển dòng Collector lớn (I_C), theo công thức: $I_C = h_{FE} \times I_B$.

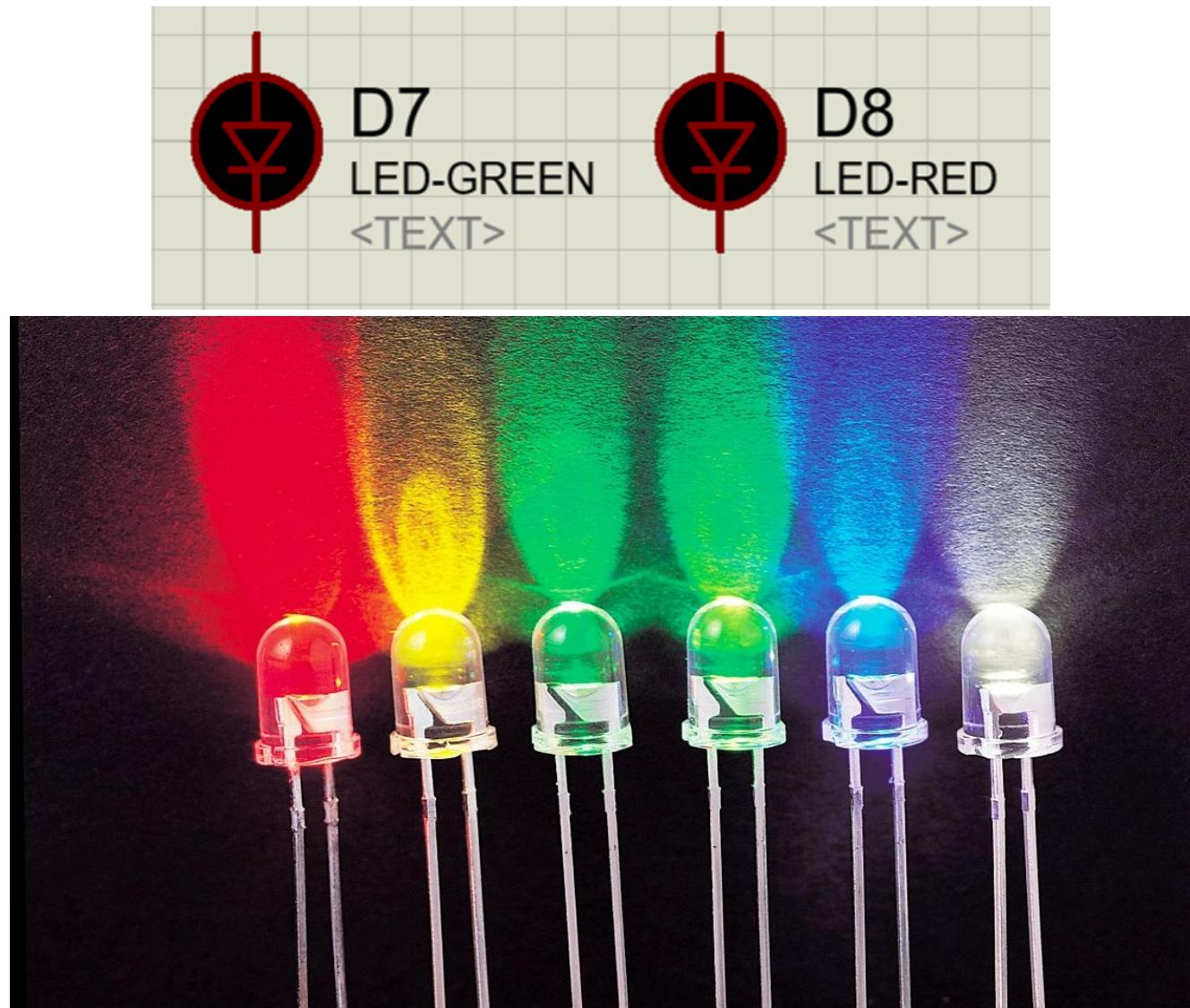
2. Chế độ hoạt động:

- Chế độ cắt (Cut-off): Khi không có dòng Base ($I_B = 0$), transistor không dẫn, và dòng Collector-Emitter (I_{CE}) bằng 0.
- Chế độ bão hòa (Saturation): Khi dòng Base đủ lớn, transistor dẫn hoàn toàn, cho phép dòng điện lớn chạy từ Collector đến Emitter với điện áp $V_{CE(sat)}$ thấp.

- Chế độ tuyển tính (Active): Transistor hoạt động như một bộ khuếch đại, với dòng Collector tỷ lệ với dòng Base.
3. Ứng dụng công tắc:
- TIP122 thường được sử dụng như một công tắc điện tử để điều khiển tải (động cơ, đèn, rơ-le) bằng tín hiệu điều khiển từ vi điều khiển hoặc mạch logic.

2.1.8 LED

Tổng quan



Hình 2.1.8 LED

LED (viết tắt của Light Emitting Diode – Đèn LED) là một loại đèn bán dẫn có khả năng phát ra ánh sáng khi được phân cực thuận. Đây là linh kiện điện tử phổ biến trong mọi lĩnh vực từ điện tử dân dụng đến công nghiệp, y tế, giao thông, chiếu sáng, hiển thị kỹ thuật số,...

LED ra đời vào những năm 1960, ban đầu chỉ có màu đỏ, công suất thấp. Qua nhiều thập kỷ phát triển, ngày nay LED đã trở thành giải pháp chiếu sáng hiệu quả, tiết kiệm năng lượng và bền vững.

Thông số kỹ thuật

Thông số kỹ thuật	Giá trị điển hình
Điện áp hoạt động (VF)	1.8 – 3.3V (tùy màu)
Dòng điện tối ưu (IF)	5mA – 20mA
Công suất tiêu thụ	~0.1W (LED thường), vài W (LED công suất)
Tuổi thọ trung bình	30.000 – 100.000 giờ
Góc phát sáng	20° – 160°
Loại ánh sáng	Đơn sắc, RGB, trắng ấm, trắng lạnh, ...

Bảng 2.1.8a Thông số kỹ thuật LED

Cấu tạo

Đèn LED có cấu tạo đơn giản nhưng được tối ưu hóa để phát ra ánh sáng hiệu quả. Các thành phần chính bao gồm:

1. Chip bán dẫn (Semiconductor Chip):
 - Là bộ phận cốt lõi, được làm từ các vật liệu bán dẫn như Gallium Arsenide (GaAs), Gallium Phosphide (GaP) hoặc Gallium Nitride (GaN).
 - Chip bao gồm hai vùng bán dẫn: vùng P (dương) và vùng N (âm), tạo thành tiếp giáp PN.
2. Lớp phát quang (Active Layer):
 - Nằm ở tiếp giáp PN, nơi xảy ra hiện tượng phát quang khi các electron tái hợp với lỗ trống (holes), giải phóng năng lượng dưới dạng photon (ánh sáng).

3. Vỏ bọc (Encapsulation):

- Thường làm từ nhựa epoxy hoặc silicon trong suốt để bảo vệ chíp bán dẫn và tăng cường phân tán ánh sáng.
- Một số LED có thấu kính tích hợp để định hướng hoặc tập trung ánh sáng.

4. Chân dẫn (Leads):

- Bao gồm hai chân: Anode (dương) và Cathode (âm), được kết nối với chíp bán dẫn thông qua dây nối mảnh (bonding wires).
- Chân Cathode thường được nhận biết bằng chân ngắn hơn hoặc có ký hiệu (vạch hoặc mặt phẳng) trên thân LED.

5. Tản nhiệt (Heat Sink):

- Trong các LED công suất cao, một bộ phận tản nhiệt (thường là tấm kim loại hoặc vật liệu dẫn nhiệt) được sử dụng để tản nhiệt, đảm bảo hiệu suất và tuổi thọ.

Nguyên lý hoạt động

Đèn LED hoạt động dựa trên hiện tượng phát quang điện (electroluminescence) trong vật liệu bán dẫn:

1. Tiếp giáp PN:

- Khi điện áp thuận được áp vào (Anode nối với cực dương, Cathode nối với cực âm), dòng điện chạy qua tiếp giáp PN.
- Các electron từ vùng N di chuyển vào vùng P và tái hợp với các lỗ trống (holes).

2. Phát quang:

- Quá trình tái hợp electron và lỗ trống giải phóng năng lượng dưới dạng photon (ánh sáng).
- Bước sóng của ánh sáng (màu sắc) phụ thuộc vào vật liệu bán dẫn và năng lượng vùng cấm (bandgap). Ví dụ:
 - GaAs: Ánh sáng đỏ hoặc hồng ngoại.
 - GaN: Ánh sáng xanh hoặc trắng.
 - GaP: Ánh sáng xanh lá hoặc vàng.

3. Điều khiển dòng điện:

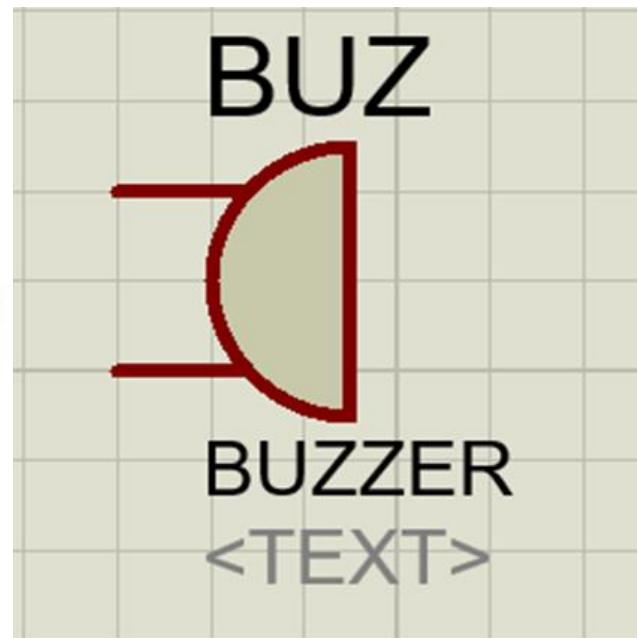
- LED yêu cầu dòng điện giới hạn (thường từ 10mA đến 20mA cho LED thông thường, hoặc cao hơn cho LED công suất cao) để tránh hư hỏng. Điều này thường được thực hiện bằng điện trở nối tiếp hoặc mạch điều khiển dòng.

Bảng 2.1.8b So sánh LED với các loại đèn truyền thống

Tiêu chí	LED	Bóng sợi đốt	Đèn huỳnh quang
Hiệu suất sáng	Cao (100 lm/W)	Thấp (10-15 lm/W)	Trung bình (60 lm/W)
Tiêu thụ điện	Rất thấp	Cao	Trung bình
Tuổi thọ	> 30.000 giờ	~1.000 giờ	~5.000 – 10.000 giờ
Độ bền cơ học	Cao	Dễ vỡ	Dễ vỡ
Tác hại môi trường	Không	Không	Có thủy ngân

2.1.9 Buzzer

Tổng quan



Hình 2.1.9 Buzzer

Buzzer, hay còn gọi là còi báo, là một linh kiện điện tử được sử dụng để tạo ra âm thanh nhằm cảnh báo, thông báo hoặc cung cấp tín hiệu âm thanh trong các hệ thống điện tử. Với thiết kế đơn giản, giá thành thấp và dễ sử dụng, buzzer được ứng dụng rộng rãi trong nhiều lĩnh vực, từ thiết bị gia dụng, điện tử tiêu dùng đến các hệ thống công nghiệp và y tế. Buzzer có thể tạo ra các âm thanh đơn giản (như tiếng bip) hoặc phức tạp hơn (như giao điệu), tùy thuộc vào loại và cách điều khiển.

Phân loại

Buzzer được phân loại dựa trên cơ chế hoạt động và cách điều khiển:

1. Theo cơ chế hoạt động:
 - Buzzer điện từ: Sử dụng cuộn dây và nam châm, phù hợp cho âm thanh lớn, tần số thấp.
 - Buzzer áp điện: Sử dụng tinh thể áp điện, nhỏ gọn, hiệu suất cao, phù hợp cho tần số cao.
2. Theo cách điều khiển:
 - Buzzer chủ động (Active Buzzer): Tự tạo âm thanh khi cấp nguồn, thường dùng trong báo động đơn giản.
 - Buzzer thụ động (Passive Buzzer): Yêu cầu tín hiệu điều khiển bên ngoài, linh hoạt hơn trong việc tạo giao điệu.
3. Theo ứng dụng:
 - Buzzer chỉ thị: Nhỏ gọn, dùng trong thiết bị điện tử (máy tính, đồng hồ báo thức).
 - Buzzer công nghiệp: Cường độ âm thanh lớn, dùng trong hệ thống báo động công nghiệp hoặc ô tô.

Cấu tạo

Buzzer có cấu tạo cơ bản bao gồm các thành phần sau, tùy thuộc vào loại cụ thể:

1. Màng rung (Diaphragm):
 - Là bộ phận chính tạo ra âm thanh, thường làm từ kim loại mỏng hoặc vật liệu composite.
 - Màng rung dao động khi có tín hiệu điện, tạo ra sóng âm.

2. Cuộn dây (Coil) và nam châm:

- Trong buzzer điện từ (electromagnetic buzzer), cuộn dây được cấp điện tạo ra từ trường, làm màng rung dao động dưới tác động của nam châm.
- Trong buzzer áp điện (piezoelectric buzzer), một tinh thể áp điện (piezoelectric crystal) được sử dụng để tạo dao động cơ học.

3. Vỏ bọc (Housing):

- Thường làm từ nhựa hoặc kim loại, bảo vệ các bộ phận bên trong và giúp khuếch đại âm thanh.
- Một số buzzer có lỗ thoát âm để tối ưu hóa âm thanh phát ra.

4. Chân kết nối (Leads):

- Bao gồm hai chân (dương và âm) để kết nối với mạch điện.
- Một số buzzer tích hợp mạch điều khiển, yêu cầu thêm chân điều khiển tín hiệu.

5. Mạch điều khiển (nếu có):

- Trong các buzzer chủ động (active buzzer), mạch dao động tích hợp tạo ra âm thanh khi được cấp nguồn.
- Trong buzzer thụ động (passive buzzer), cần tín hiệu bên ngoài (như sóng vuông) để tạo âm thanh.

Bảng 2.1.9 Thông số kỹ thuật Buzzer

Thông số	Giá trị điển hình
Điện áp hoạt động	3V – 12V (thường dùng 5V)
Dòng tiêu thụ	10 – 30mA
Tần số phát âm	~2kHz (buzzer chủ động), có thể thay đổi (buzzer thụ động)
Cường độ âm thanh	80 – 100 dB @10cm
Đường kính phổ biến	9mm, 12mm, 15mm, 23mm,...
Kiểu vỏ	Tròn (có lỗ), chân cắm (pin), dán (SMD)

Nguyên lý hoạt động

Nguyên lý hoạt động của buzzer phụ thuộc vào loại buzzer, nhưng cơ bản dựa trên việc chuyển đổi tín hiệu điện thành dao động cơ học để tạo ra âm thanh:

1. Buzzer điện từ (Electromagnetic Buzzer):
 - Khi dòng điện chạy qua cuộn dây, từ trường được tạo ra làm màng rung dao động gần nam châm.
 - Tần số dao động của màng quyết định cao độ âm thanh (thường từ 2kHz đến 4kHz).
 - Buzzer điện từ thường được sử dụng trong các ứng dụng yêu cầu âm thanh lớn và đơn giản.
2. Buzzer áp điện (Piezoelectric Buzzer):
 - Tinh thể áp điện co giãn khi được áp điện áp, tạo ra dao động cơ học trên màng rung.
 - Tín hiệu điện xoay chiều (AC) hoặc sóng vuông được sử dụng để điều khiển tần số dao động, từ đó tạo ra âm thanh.
 - Buzzer áp điện có hiệu suất cao hơn và thường nhỏ gọn hơn buzzer điện từ.
3. Buzzer chủ động (Active Buzzer):
 - Có mạch dao động tích hợp, chỉ cần cấp nguồn DC (thường 3-12V) là tạo ra âm thanh.
 - Âm thanh thường là một tần số cố định (ví dụ: tiếng bíp).
4. Buzzer thụ động (Passive Buzzer):
 - Yêu cầu tín hiệu điều khiển bên ngoài (thường là sóng vuông với tần số cụ thể) để tạo âm thanh.
 - Có thể tạo ra nhiều tần số hoặc giai điệu khác nhau, tùy thuộc vào tín hiệu đầu vào.

2.2 PHẦN MỀM KEIL C

Tổng quan

Keil C, hay chính xác hơn là Keil µVision, là một môi trường phát triển tích hợp (Integrated Development Environment - IDE) được phát triển bởi Keil Software (hiện thuộc sở hữu của Arm Limited). Phần mềm này được thiết kế để hỗ trợ lập trình và phát triển phần mềm nhúng cho các vi điều khiển dựa trên kiến trúc 8051, Arm Cortex-M, và một số dòng vi điều khiển khác. Keil C cung cấp các công cụ mạnh mẽ để viết mã, biên dịch, gỡ lỗi và nạp chương trình vào vi điều khiển, là lựa chọn hàng đầu cho các kỹ sư và nhà phát triển hệ thống nhúng.

Tính năng và cấu trúc của Keil C

1. Các thành phần chính

Keil µVision là một IDE toàn diện với các thành phần chính sau:

- Trình soạn thảo mã nguồn (Source Code Editor):
- Hỗ trợ viết mã bằng ngôn ngữ C/C++ và Assembly.
- Tích hợp tính năng tô màu cú pháp (syntax highlighting), tự động thụt lề và gợi ý mã để tăng hiệu quả lập trình.
- Trình biên dịch (Compiler):
 - Bao gồm trình biên dịch C/C++ cho các kiến trúc vi điều khiển như 8051, Arm Cortex-M (Cortex-M0, M3, M4, M7), và C166.
 - Tối ưu hóa mã để giảm kích thước chương trình và tăng hiệu suất thực thi.
- Trình liên kết (Linker):
 - Tạo tệp thực thi (thường ở định dạng HEX hoặc BIN) để nạp vào vi điều khiển.
 - Hỗ trợ quản lý bộ nhớ và ánh xạ địa chỉ.
- Trình gỡ lỗi (Debugger):
 - Cung cấp khả năng gỡ lỗi mạnh mẽ với các công cụ như breakpoint, step-by-step execution, và xem giá trị biến trong thời gian thực.
 - Hỗ trợ gỡ lỗi thông qua các giao diện như JTAG, SWD hoặc các trình mô phỏng phần cứng.
- Trình mô phỏng (Simulator):
 - Cho phép mô phỏng hoạt động của vi điều khiển mà không cần phần cứng thực tế.

- Hỗ trợ kiểm tra logic chương trình và tối ưu hóa trước khi nạp vào thiết bị.
- Quản lý dự án (Project Management):
- Hỗ trợ tổ chức dự án với cấu trúc thư mục, quản lý tệp mã nguồn và thư viện.
- Tích hợp với các công cụ bên thứ ba như trình nạp chương trình (programmer) và hệ thống kiểm soát phiên bản.
- Hỗ trợ phần cứng:
- Tương thích với các công cụ lập trình và gỡ lỗi như ULINK, J-Link, ST-Link, và các trình nạp chương trình của nhà sản xuất vi điều khiển.

2. Các phiên bản

Keil μVision có các phiên bản chính như sau:

- Keil μVision 5: Phiên bản phổ biến nhất hiện nay, hỗ trợ nhiều dòng vi điều khiển và tích hợp các công cụ hiện đại.
- Keil C51: Dành riêng cho vi điều khiển 8051 và các biến thể.
- Keil MDK (Microcontroller Development Kit): Tập trung vào các vi điều khiển Arm Cortex-M, bao gồm thư viện CMSIS và hỗ trợ RTOS (Real-Time Operating System).

3. Ngôn ngữ lập trình

- **Ngôn ngữ chính:** C/C++ (Keil C là trình biên dịch C tối ưu hóa cho vi điều khiển).
- **Assembly:** Hỗ trợ lập trình cấp thấp để tối ưu hóa hiệu suất và điều khiển phần cứng trực tiếp.
- **Hỗ trợ RTOS:** Tích hợp RTX RTOS (Real-Time Executive) của Arm, giúp quản lý tác vụ trong các ứng dụng nhúng phức tạp.

Nguyên lý hoạt động

Keil μVision hoạt động theo quy trình phát triển phần mềm nhúng tiêu chuẩn:

1. Viết mã nguồn:
 - Người dùng viết mã bằng C/C++ hoặc Assembly trong trình soạn thảo của Keil μVision.

- Mã được tổ chức thành các dự án với cấu hình cụ thể cho vi điều khiển mục tiêu.

2. Biên dịch và liên kết:

- Trình biên dịch chuyển mã nguồn thành mã máy (machine code).
- Trình liên kết ánh xạ mã máy vào bộ nhớ của vi điều khiển và tạo tệp thực thi (HEX/BIN).

3. Mô phỏng hoặc gỡ lỗi:

- Người dùng có thể mô phỏng chương trình trên trình mô phỏng tích hợp để kiểm tra logic.
- Gỡ lỗi được thực hiện bằng cách kết nối với phần cứng qua giao diện như JTAG hoặc SWD.

4. Nạp chương trình:

- Tệp thực thi được nạp vào vi điều khiển thông qua trình nạp chương trình (programmer) như ULINK hoặc ST-Link.

5. Kiểm tra và tối ưu hóa:

- Dựa trên kết quả gỡ lỗi, người dùng chỉnh sửa mã nguồn để tối ưu hóa hiệu suất hoặc sửa lỗi.

Thành phần	Chức năng chính
μ Vision IDE	Giao diện lập trình tích hợp, viết mã, biên dịch và mô phỏng
C Compiler (C51)	Biên dịch chương trình viết bằng C thành mã máy cho vi điều khiển
Assembler (A51)	Hỗ trợ viết và biên dịch chương trình Assembly
Linker	Liên kết các module mã đã biên dịch thành tập tin .HEX
Debugger	Mô phỏng, kiểm tra, gỡ lỗi chương trình

2.3 PHẦN MỀM PROTEUS

Proteus là một phần mềm thiết kế và mô phỏng mạch điện tử được phát triển bởi Labcenter Electronics, một công ty có trụ sở tại Vương quốc Anh. Được ra mắt lần đầu vào những năm

1980, Proteus đã trở thành một trong những công cụ phổ biến nhất cho các kỹ sư điện tử, nhà phát triển hệ thống nhúng và sinh viên để thiết kế, mô phỏng và kiểm tra các mạch điện tử mà không cần phần cứng thực tế. Phần mềm này tích hợp hai module chính: ISIS (dùng cho thiết kế và mô phỏng mạch) và ARES (dùng cho thiết kế bảng mạch in - PCB).

Tính năng và cấu trúc của Proteus

1. Các thành phần chính

Proteus bao gồm hai module chính và một số công cụ hỗ trợ:

- ISIS (Intelligent Schematic Input System):
 - Module thiết kế sơ đồ mạch (schematic design) và mô phỏng.
 - Cho phép vẽ sơ đồ mạch điện tử với giao diện kéo-thả thân thiện.
 - Hỗ trợ mô phỏng mạch analog, digital và các hệ thống nhúng với vi điều khiển.
 - Tích hợp thư viện linh kiện phong phú, bao gồm các linh kiện như resistor, capacitor, transistor, IC, LED, buzzer, và vi điều khiển (8051, PIC, AVR, ARM).
- ARES (Advanced Routing and Editing Software):
 - Module thiết kế bảng mạch in (PCB layout).
 - Hỗ trợ định tuyến tự động (auto-routing) và thủ công (manual routing) để tạo layout PCB.
 - Tích hợp tính năng kiểm tra lỗi thiết kế (Design Rule Check - DRC) và xuất file sản xuất (Gerber, BOM).
- VSM (Virtual System Modelling):
 - Công cụ mô phỏng mạnh mẽ trong ISIS, cho phép mô phỏng hành vi của mạch với các linh kiện thực tế và vi điều khiển.
 - Hỗ trợ mô phỏng thời gian thực, bao gồm cả các giao thức như I2C, SPI, UART.
- Thư viện linh kiện:
 - Cung cấp hàng nghìn linh kiện điện tử, từ linh kiện thụ động (điện trở, tụ điện) đến linh kiện tích hợp (IC, transistor, vi điều khiển).
 - Cho phép người dùng tạo linh kiện tùy chỉnh nếu cần.

- Hỗ trợ vi điều khiển:
- Hỗ trợ lập trình và mô phỏng vi điều khiển phổ biến như 8051, PIC, AVR, ARM Cortex-M, và Arduino.
- Tích hợp với các trình biên dịch bên ngoài như Keil C, MPLAB, hoặc IAR để nạp mã HEX vào vi điều khiển trong môi trường mô phỏng.
- Công cụ phân tích:
- Cung cấp các công cụ như oscilloscope ảo, logic analyzer, và biểu đồ điện áp/dòng điện để phân tích hành vi mạch.

2. Các phiên bản

Proteus có các phiên bản chính như sau:

- Proteus Professional: Phiên bản đầy đủ với tất cả tính năng của ISIS và ARES, phù hợp cho các kỹ sư chuyên nghiệp.
- Proteus Lite: Phiên bản giới hạn, phù hợp cho sinh viên hoặc các dự án nhỏ.
- Proteus PCB Design: Tập trung vào thiết kế PCB với các cấp độ khác nhau (Level 1, Level 2, Level 3) tùy thuộc vào số lượng chân linh kiện.
- Proteus VSM: Tích hợp mô phỏng vi điều khiển và hệ thống nhúng.

3. Hỗ trợ lập trình

- **Ngôn ngữ lập trình:** Hỗ trợ các tệp HEX hoặc ELF được tạo từ các trình biên dịch như Keil C, MPLAB, IAR, hoặc GCC.
- **Mô phỏng vi điều khiển:** Cho phép mô phỏng hành vi của vi điều khiển với mã chương trình, hỗ trợ các giao thức giao tiếp và ngoại vi (ADC, PWM, Timer).
- **Nguyên lý hoạt động**

Proteus hoạt động theo quy trình thiết kế và mô phỏng mạch điện tử tiêu chuẩn:

1. Thiết kế sơ đồ mạch:
- Người dùng sử dụng module ISIS để vẽ sơ đồ mạch bằng cách kéo-thả linh kiện từ thư viện và kết nối chúng bằng dây dẫn ảo.
- Có thể thêm vi điều khiển và nạp tệp HEX để mô phỏng hành vi lập trình.

2. Mô phỏng mạch:

- Sử dụng VSM để mô phỏng hành vi của mạch trong thời gian thực.
- Người dùng có thể đặt các điểm đo (probe), sử dụng oscilloscope ảo hoặc logic analyzer để kiểm tra tín hiệu.

3. Thiết kế PCB:

- Chuyển sơ đồ mạch từ ISIS sang ARES để thiết kế layout PCB.
- Tùy chỉnh định tuyến và kiểm tra lỗi thiết kế để đảm bảo bảng mạch phù hợp với sản xuất.

4. Xuất và sản xuất:

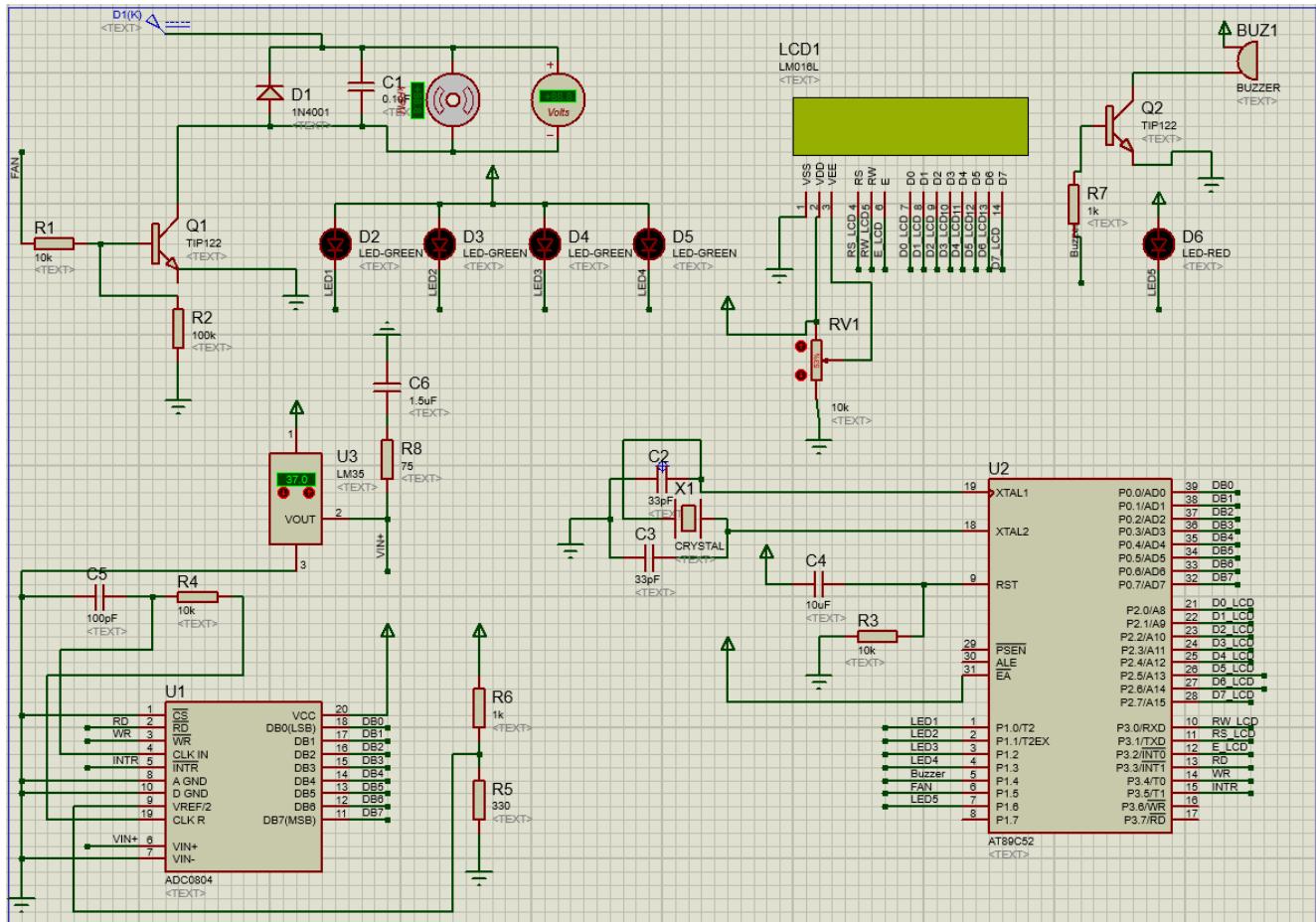
- Xuất tệp Gerber, BOM (Bill of Materials) hoặc các định dạng khác để sản xuất PCB thực tế.
- Kiểm tra lại toàn bộ thiết kế để đảm bảo không có lỗi trước khi sản xuất.

Đặc điểm kỹ thuật tiêu biểu

- Hệ điều hành hỗ trợ: Windows XP/7/8/10/11 (32-bit và 64-bit).
- Vi điều khiển hỗ trợ: 8051, PIC, AVR, ARM Cortex-M, Arduino, và nhiều dòng khác.
- Linh kiện hỗ trợ: Hơn 10.000 linh kiện analog, digital, và nhúng.
- Định dạng xuất PCB: Gerber, ODB++, DXF, PDF.
- Giao thức mô phỏng: I2C, SPI, UART, CAN, USB, v.v.
- Yêu cầu hệ thống:
 - RAM: Tối thiểu 4GB (khuyến nghị 8GB).
 - CPU: Intel Core i3 hoặc cao hơn.
 - Dung lượng ổ cứng: Khoảng 1GB cho cài đặt.

CHƯƠNG 3 THỰC HIỆN CHI TIẾT

3.0. Sơ đồ mạch mô phỏng trên Proteus



Hình 3.0 Sơ đồ mô phỏng mạch trên Proteus

3.1. Thực hiện LM35

3.1.1. Vai trò của LM35

LM35 là cảm biến nhiệt độ tuyến tính – nó chuyển đổi nhiệt độ thành điện áp analog.

Mỗi độ C tương ứng với 10mV, ví dụ:

- $25^{\circ}\text{C} \rightarrow 250\text{mV}$
 - $30^{\circ}\text{C} \rightarrow 300\text{mV}$
- Cảm biến này là trung tâm của hệ thống điều khiển vì:
- Cung cấp tín hiệu đầu vào cho bộ ADC0804

- Làm căn cứ để điều chỉnh tốc độ quạt, bật LED cảnh báo
- Dữ liệu nhiệt độ được hiển thị lên LCD

3.1.2. Kết nối chân của LM35

Bảng 3.1 Kết nối của LM35 trong mạch

Chân	Tên chân	Chức năng	Kết nối trong mạch
1	VCC	Nguồn dương	Nối Vcc 5V (cùng nguồn với vi điều khiển)
2	Vout	Xuất tín hiệu điện áp tương ứng nhiệt độ	Nối vào chân Vin+ của ADC0804 và rẽ xuống GND qua một mạch RC (lọc nhiễu) (R8 (75Ω) và tụ điện C6 ($1.5\mu F$))
3	GND	Nguồn âm	Nối GND (mass) của toàn mạch

Chức năng của điện trở R8 (75Ω) và tụ điện C6 ($1.5\mu F$)

1. Lọc nhiễu tín hiệu đầu ra của LM35 (Low-pass filter):

- Khi LM35 xuất tín hiệu điện áp tương ứng với nhiệt độ ra chân 2 (VOUT), tín hiệu này có thể bị nhiễu do dao động nguồn, mạch ADC, hay nhiễu từ các thiết bị khác.
- Việc mắc thêm tụ C6 nối xuống GND tạo thành một bộ lọc thông thấp (low-pass filter) cùng với điện trở R8.
- Bộ lọc này giúp loại bỏ nhiễu tần số cao, làm cho tín hiệu nhiệt độ ổn định hơn trước khi đưa vào ADC (bộ chuyển đổi tương tự – số).

2. Chống dao động và tăng độ ổn định cho LM35:

- Theo datasheet của LM35, nếu dây nối dài hoặc môi trường có nhiều nhiễu, nên thêm tụ gồm vài μF (thường là $1\mu F$ đến $10\mu F$) từ VOUT về GND để cải thiện ổn định.
- Việc thêm cả điện trở nhỏ (như 75Ω) làm chậm tốc độ thay đổi tín hiệu, giúp giảm bớt nhiễu tần số cao mà không ảnh hưởng nhiều đến tín hiệu nhiệt độ (do LM35 phản hồi khá chậm).

3.2. Thực hiện LCD 16x2

3.2.1. Vai trò LCD 16x2

Màn hình LCD (Liquid Crystal Display) 16x2 là thiết bị hiển thị chính của hệ thống, có vai trò cung cấp thông tin trực quan cho người dùng về tình trạng nhiệt độ, trạng thái quạt, và cảnh báo.

Hiển thị thông tin nhiệt độ

- LCD nhận dữ liệu từ vi điều khiển AT89S52 thông qua các chân dữ liệu và chân điều khiển.
- Dữ liệu nhiệt độ sau khi xử lý được chuyển sang dạng ký tự ASCII và truyền đến LCD để hiển thị.
- LCD có thể hiển thị 2 dòng, mỗi dòng 16 ký tự. Trong hệ thống này, các thông tin được hiển thị bao gồm:
 - Giá trị nhiệt độ thực tế (ví dụ: "TEMP: 36°C")
 - Cảnh báo nếu nhiệt độ vượt ngưỡng (ví dụ: "ALERT!" cùng với nhiệt độ tương ứng)

Vai trò điều khiển hiển thị

- Vi điều khiển sử dụng **Port 2 (P2.0 – P2.7)** để gửi dữ liệu 8-bit đến LCD.
- Các tín hiệu điều khiển RS, RW, EN được kết nối như sau:
 - P3.1 (RS): Lựa chọn chế độ (lệnh hoặc dữ liệu).
 - P3.0 (RW): Ghi (0) hoặc đọc (1) dữ liệu từ LCD.
 - P3.2 (EN): Tín hiệu cho phép, kích hoạt việc nhận lệnh hoặc dữ liệu.
- Mỗi khi LCD cần cập nhật thông tin mới, vi điều khiển gửi lệnh/xung theo trình tự logic qua các chân điều khiển để đảm bảo LCD hoạt động ổn định.

LCD giao tiếp ở chế độ **8-bit** và dùng các chân điều khiển để hiển thị dữ liệu. Theo quy ước LCD 16x2 chuẩn:

Bảng 3.2 Kết nối của LCD16x2 trong mạch

Chân số	Tên chân LCD	Chức năng	Kết nối trong mạch
1	VSS	Nối đất (0V)	Nối GND
2	VDD	Cấp nguồn dương (+5V)	Nối VCC
3	V0(VEE)	Điều chỉnh độ tương phản màn hình	Nối vào chiết áp RV1 (10k) để chỉnh độ tương phản LCD
4	RS	Chọn thanh ghi (0: Lệnh, 1: Dữ liệu)	Nối với P2.7 của vi điều khiển AT89C52
5	RW	Chọn chế độ (0: ghi, 1: đọc)	Nối GND (chế độ ghi luôn)
6	EN	Tín hiệu kích hoạt (Enable) để đọc dữ liệu/lệnh	Nối với P2.6 của vi điều khiển
7–14	D0–D7	8 chân dữ liệu song song	Nối với P0.0 – P0.7 của vi điều khiển (sử dụng truyền song song 8 bit)
15	LED+ (A)	Cực dương của đèn nền	Nối VCC qua điện trở (hoặc trực tiếp nếu không cần giảm sáng)
16	LED– (K)	Cực âm của đèn nền	Nối GND

3.3. Thực hiện ADC0804

ADC0804 là vi mạch chuyển đổi tín hiệu tương tự (analog) sang tín hiệu số (digital), giúp vi điều khiển AT89S52 có thể xử lý dữ liệu từ cảm biến nhiệt độ LM35.

Chuyển đổi tín hiệu analog sang số

- Cảm biến LM35 xuất ra điện áp analog tỷ lệ tuyến tính với nhiệt độ ($10\text{mV}/^\circ\text{C}$).
- ADC0804 nhận tín hiệu analog từ LM35 thông qua chân **Vin(+)** và thực hiện quá trình chuyển đổi sang giá trị nhị phân 8 bit.
- Giá trị nhị phân được gửi đến vi điều khiển qua Port 0 (P0.0 – P0.7).

Quá trình giao tiếp với vi điều khiển

- Vi điều khiển sử dụng các chân của Port 3 để điều khiển hoạt động của ADC0804:

- **P3.4 (WR)**: Khi xuống mức thấp, bắt đầu quá trình chuyển đổi analog sang digital.
- **P3.5 (INTR)**: Khi ADC chuyển đổi xong, chân này xuống mức thấp để báo hiệu dữ liệu đã sẵn sàng.
- **P3.3 (RD)**: Sau khi nhận tín hiệu INTR, vi điều khiển kéo RD xuống thấp để đọc dữ liệu từ ADC.
- Sau khi đọc xong, P3.3 được kéo lên lại để kết thúc chu kỳ đọc.

Cấu hình và kết nối

- ADC0804 cần nguồn 5V để hoạt động ổn định, nối VCC và GND phù hợp.
- Một tụ điện (thường 150pF) và điện trở kéo (10kΩ) có thể được nối tại các chân thời gian để điều chỉnh tốc độ chuyển đổi.
- Tốc độ chuyển đổi của ADC phụ thuộc vào xung clock ngoài, có thể dùng RC hoặc thạch anh riêng cho ADC nếu cần tốc độ cao hơn.

Chân	Tên chân	Mô tả chức năng	Chức năng cụ thể trong mạch
1	CS	Chip Select – bật ADC	Nối GND để luôn bật ADC , không cần kích hoạt nhiều lần.
2	R[̄]D	Read – cho phép đọc dữ liệu	Nối tới vi điều khiển P3.3 , khi MCU kéo xuống LOW thì dữ liệu được đọc từ ADC.
3	W[̄]R	Write – bắt đầu chuyển đổi	Nối tới vi điều khiển P3.4 , khi MCU kéo xuống LOW thì ADC bắt đầu chuyển đổi analog -> digital.
4	CLK IN	Clock nội	Được nối với chân 19 (CLK R) để tạo dao động nội bằng RC.
5	INTR	Interrupt – báo hiệu ADC đã xong	Nối tới vi điều khiển P3.5 , để biết khi nào dữ liệu sẵn sàng để đọc.
6	V_{in+}	Tín hiệu analog dương	Nối đến chân 2 của LM35 – xuất tín hiệu điện áp tỷ lệ nhiệt độ (10mV/°C).
7	V_{in-}	Tín hiệu analog âm (thường GND)	Nối đất (GND), tạo hiệu điện thế với V _{in+} .

8	AGND	Đất analog	Nối GND. Cần nối đúng để ADC hoạt động chính xác.
9	VREF/2	Điện áp tham chiếu/2	Mặc định nếu không nối gì , dải đo sẽ là 0–5V (nếu VCC = 5V).
10	CLK R	Tạo dao động RC nội	Nối với R7 (10k) và C6 (150pF) để tạo xung clock ~ 640kHz.
11–18	DB7–DB0	8-bit dữ liệu đầu ra digital	Nối tới PORT0 (P0.0–P0.7) của vi điều khiển để đọc dữ liệu.
20	VCC	Nguồn cấp 5V	Nối +5V để cấp nguồn cho ADC0804.

Bảng 3.3 Kết nối của ADC0804 trong mạch

3.4. Thực hiện AT89S52

Trong dự án này, vi điều khiển chính được sử dụng là AT89S52, thuộc họ 8051 của hãng Atmel. Đây là vi điều khiển 8-bit có kiến trúc Harvard, gồm 40 chân, với bộ nhớ Flash 8KB, RAM 256 byte, và hỗ trợ 2 Timer/Counter, 4 cổng I/O và các tính năng ngắt.

Tuy nhiên, trong quá trình mô phỏng trên phần mềm Proteus, do hạn chế về thư viện linh kiện, nhóm sử dụng AT89C52 để thay thế trong sơ đồ mô phỏng. Hai vi điều khiển này hoàn toàn tương thích chân và mã lệnh, do đó việc thay thế không làm thay đổi chức năng hay kết quả mô phỏng.

So sánh:

Đặc điểm	AT89S52	AT89C52
Bộ nhớ Flash	8 KB	8 KB
RAM nội	256 byte	256 byte
Tốc độ tối đa	33 MHz	24 MHz
Giao tiếp ISP	Có (In-System)	Không có
Tương thích code	✓	✓

Bảng 3.4.a So sánh AT89S52 và AT89C52

Như vậy, trong thực tế nhóm dùng AT89S52 để có thể nạp chương trình dễ dàng qua giao tiếp ISP nên dùng AT89C52 trong Proteus là hoàn toàn phù hợp cho mô phỏng phần mềm và phần cứng logic.

3.4.1. Vai trò của vi điều khiển AT89S52 trong hệ thống của đề tài

Vi điều khiển AT89S52 đóng vai trò trung tâm trong hệ thống đo và điều khiển nhiệt độ. Nó chịu trách nhiệm thực hiện tất cả các chức năng xử lý tín hiệu, điều khiển thiết bị đầu ra và hiển thị thông tin. Cụ thể:

Nhận dữ liệu từ cảm biến nhiệt độ

- Cảm biến nhiệt độ LM35 xuất tín hiệu analog tuyến tính theo nhiệt độ ($10\text{mV}/^\circ\text{C}$).
- Tín hiệu analog này được đưa vào chân Vin của ADC0804 để chuyển đổi sang dạng số.
- Sau khi chuyển đổi xong, ADC0804 gửi dữ liệu số 8-bit sang Port 0 (P0.0 – P0.7) của AT89S52.

Xử lý dữ liệu nhiệt độ

- AT89S52 đọc dữ liệu từ ADC0804, sau đó thực hiện các bước:
 - So sánh mức nhiệt độ với các ngưỡng được lập trình sẵn.

- Quyết định tốc độ quạt phù hợp với từng mức nhiệt độ.
- Chuyển đổi dữ liệu nhiệt độ sang dạng ký tự để hiển thị.

Hiển thị nhiệt độ trên LCD

- Vi điều khiển sử dụng các chân P2.0 – P2.7 để truyền dữ liệu 8-bit tới LCD.
- Các tín hiệu điều khiển RS, RW, EN được xuất từ P3.0, P3.1, P3.2 tương ứng.
- Thông tin hiển thị gồm: giá trị nhiệt độ, thông báo mức quạt, cảnh báo...

Điều khiển tốc độ quạt theo nhiệt độ

- Dựa trên kết quả so sánh nhiệt độ, AT89S52 thay đổi chế độ hoạt động của Timer0 để tạo xung điều khiển PWM đơn giản.
- Chân P1.5 được sử dụng để xuất tín hiệu PWM điều khiển quạt thông qua một transistor.
- Các mức tốc độ tương ứng với các mức nhiệt độ (ví dụ: <30°C: chậm, >40°C: nhanh...).

Điều khiển đèn LED báo mức nhiệt

- Các chân P1.0 – P1.3 điều khiển 4 LED báo mức nhiệt độ từ thấp đến cao và 1 LED cảnh báo vượt ngưỡng 40°C.
- Khi nhiệt độ tăng, LED sáng trực quan theo từng mức nhiệt độ hiện tại.

Cảnh báo nhiệt độ cao

- Khi nhiệt độ vượt ngưỡng an toàn (trên 40°C):
 - Chân P1.4 điều khiển sáng LED đỏ cảnh báo.
 - Chân P1.6 điều khiển còi buzzer báo động.
 - Dòng chữ cảnh báo “ALERT!” được hiển thị trên LCD.

Điều khiển thời gian thực bằng ngắt

- Vi điều khiển sử dụng Timer0 và ngắt ngoài để tạo xung điều khiển quạt và thực hiện kiểm tra định kỳ.
- Ngắt giúp AT89S52 thực hiện tác vụ một cách chính xác theo thời gian, ví dụ:
 - Thay đổi xung PWM.
 - Đảo trạng thái chân P1.5 để điều khiển quạt quay theo chu kỳ.

3.4.2. Phân tích chân AT89S52 nối trong mạch

Chân	Tên	Mô tả chức năng	Kết nối trong mạch
40	VCC	Nguồn +5V	Nối với nguồn VCC
20	GND	Dây Mass: điểm âm nguồn (0V), tạo tham chiếu điện áp cho tất cả các chân logic của vi điều khiển.	Nối GND
18-19	XTAL1/2	Đao động ngoài (thạch anh 24MHz) tạo xung clock. Nối tụ giúp ổn định dao động, tránh nhiễu.	Nối thạch anh 24MHz và 2 tụ 33pF xuống GND
32–39	P0.0–P0.7	Cổng I/O – dữ liệu 8 bit	Nhận dữ liệu từ ADC0804 (D0–D7)
21–28	P2.0–P2.7	Cổng I/O – dữ liệu điều khiển LCD	Gửi dữ liệu 8-bit đến LCD
1	P1.0	Điều khiển LED tốc độ/quạt mức 1	Nối LED 1 (LED-GREEN)
2	P1.1	Điều khiển LED tốc độ/quạt mức 2	Nối LED 2 (LED-GREEN)
3	P1.2	Điều khiển LED tốc độ/quạt mức 3	Nối LED 3 (LED-GREEN)
4	P1.3	Điều khiển LED tốc độ/quạt mức 4	Nối LED 4 (LED-GREEN)
5	P1.4	Điều khiển còi cảnh báo	Nối còi (buzzer)
6	P1.5	Xuất xung PWM điều khiển tốc độ quạt PWM được tạo bằng Timer 0 + ngắt (CPL P1.5 trong hàm ngắt).	Nối quạt (thông qua điện trở 10kΩ với điện trở 100kΩ nối GND và TIP122, Diode 1N4001 chống dòng ngược, Tụ 0.1μF chống nhiễu)
7	P1.6	Điều khiển LED cảnh báo/quạt mức 5	Nối LED 5 (LED-RED)
8	P1.7	Không sử dụng	Không kết nối
9	RST	Reset vi điều khiển	Tụ 10μF xuống GND và điện trở 10kΩ lên VCC
10	P3.0/RXD	Điều khiển RW (Read/Write) của LCD RW = 0: Ghi dữ liệu từ vi điều khiển vào LCD.	Nối RW LCD

		RW = 1: Đọc dữ liệu từ LCD về vi điều khiển.	
11	P3.1/TXD	Điều khiển RS (Register Select) của LCD RS = 0: Gửi lệnh RS = 1: Gửi dữ liệu (ký tự để hiển thị)	Nối RS LCD
12	P3.2/INT0	Tạo xung (từ thấp lên cao rồi xuống thấp) cho phép để LCD nhận lệnh/dữ liệu từ MCU	Nối E LCD
13	P3.3/INT1	VĐK kéo P3.3 xuống mức thấp (CLR P3.3), nó kích hoạt chân RD ADC0804, cho phép đọc dữ liệu số đã chuyển đổi từ analog. Đọc xong trả lại P3.3 = 1	Nối RD ADC0804
14	P3.4/T0	Dùng làm chân WR (Write) để gửi yêu cầu bắt đầu quá trình chuyển đổi cho ADC0804.	Nối WR ADC0804
15	P3.5/T1	Dùng làm chân INTR (Interrupt) là tín hiệu báo hiệu ADC đã chuyển đổi xong.	INTR ADC0804
16	P3.6/WR	Không sử dụng	Không kết nối
17	P3.7RD	Không sử dụng	Không kết nối

Bảng 3.4.b Kết nối của AT89S52 trong mạch

3.5. Quạt và Led và Buzzer

3.5.1. Vai trò trong hệ thống

Trong hệ thống điều khiển nhiệt độ sử dụng vi điều khiển AT89S52, quạt và LED đóng vai trò là các thiết bị đầu ra, cho phép hệ thống phản hồi trực quan theo thời gian thực dựa trên nhiệt độ môi trường. Cụ thể:

- Quạt:
 - Được sử dụng để làm mát không gian khi nhiệt độ vượt quá các ngưỡng cài đặt.
 - Dựa vào giá trị nhiệt độ thu được từ cảm biến LM35 và xử lý qua ADC0804, vi điều khiển sẽ điều chỉnh tốc độ của quạt bằng cách điều khiển các chân xuất ra từ PORT 1.
 - Tốc độ quạt tăng dần theo từng mức nhiệt, giúp hệ thống làm mát hiệu quả và tiết kiệm năng lượng.
- Buzzer:

- Buzzer là một thiết bị đầu ra dạng âm thanh nhận tín hiệu từ AT89S52, được sử dụng trong hệ thống nhằm cảnh báo người dùng khi nhiệt độ vượt quá mức an toàn. Cụ thể:
- Khi nhiệt độ môi trường đo được vượt qua ngưỡng nguy hiểm ($\geq 40^{\circ}\text{C}$), hệ thống không chỉ bật LED cảnh báo mà còn kích hoạt buzzer để phát ra âm thanh cảnh báo liên tục hoặc theo chu kỳ.
- Điều này giúp cảnh báo người dùng ngay cả khi không quan sát trực tiếp hệ thống, đặc biệt trong các môi trường không thể theo dõi màn hình LCD hoặc LED.

Buzzer đóng vai trò quan trọng trong việc đảm bảo an toàn và nâng cao tính tương tác của hệ thống, đặc biệt trong các tình huống khẩn cấp về nhiệt độ.

- **LED:**

Các đèn LED cung cấp thông tin trực quan về trạng thái hoạt động của quạt và hệ thống. Cụ thể:

- LED báo mức quạt:
 - P1.0 sáng: Quạt đang hoạt động ở mức 1 (thấp nhất).
 - P1.1 sáng: Quạt hoạt động ở mức 2.
 - P1.2 sáng: Quạt hoạt động ở mức 3.
 - P1.3 sáng: Quạt hoạt động ở mức 4 (cao nhất).
- LED cảnh báo nhiệt độ cao:
 - P1.3 cùng với P1.6 sáng khi nhiệt độ vượt ngưỡng an toàn ($\geq 40^{\circ}\text{C}$), cảnh báo người dùng.

Nhờ hệ thống LED và quạt được điều khiển tự động bởi chương trình vi điều khiển, người dùng có thể dễ dàng giám sát tình trạng vận hành của hệ thống và môi trường xung quanh một cách trực quan và hiệu quả.

3.5.2. Kết nối trong hệ thống

Chân vi điều khiển	Ký hiệu chân	Vai trò	Ghi chú
1	P1.0	LED báo mức quạt 1	Sáng khi quạt ở tốc độ thấp nhất
2	P1.1	LED báo mức quạt 2	Sáng khi quạt ở tốc độ trung bình thấp
3	P1.2	LED báo mức quạt 3	Sáng khi quạt ở tốc độ trung bình cao
4	P1.3	LED báo mức quạt 4	Sáng khi quạt ở tốc độ cao
5	P1.4	Buzzer	Điều khiển còi cảnh báo
6	P1.5	Điều khiển quạt chạy tương ứng với từng mức	Xuất xung PWM điều khiển tốc độ quạt PWM được tạo bằng Timer 0 + ngắn
7	P1.6	LED 5 cảnh báo với mức quạt 5	Sáng khi quạt ở tốc độ cao nhất cùng với Buzzer cảnh báo

Bảng 3.5 Kết nối của Led, quạt và Buzzer trong mạch

3.6. Mạch cung cấp nguồn, Reset và dao động hệ thống

3.6.1. Mạch cung cấp nguồn

Hệ thống sử dụng **nguồn một chiều 5V DC** để cấp điện cho mạch, bao gồm: vi điều khiển AT89S52, LCD 16x2, ADC0804, cảm biến nhiệt độ LM35, LED hiển thị, buzzer, riêng quạt DC dùng **nguồn 12V** để cấp điện.

Trong mạch Proteus mô phỏng:

- **Nguồn 5V** được nối trực tiếp qua USBasp và nối trực tiếp đến mạch:
 - Với Chân 40 (VCC) của AT89S52.
 - Chân VCC của các linh kiện khác như ADC0804, LCD, LED, buzzer, và cảm biến LM35.
- **Nguồn 12V** được cấp từ bộ adapter từ bên ngoài vào cấp điện cho quạt DC.

- Chân GND (chân 20) của AT89S52 và các linh kiện khác đều được nối chung về chung mass (GND).

3.6.2. Mạch cung cấp nguồn

Mạch Reset bảo đảm cho vi điều khiển khởi động đúng sau khi cấp nguồn hoặc khi cần reset thủ công.

Cấu hình trong mạch:

- **Chân 9 (RST)** được kết nối như sau:
 - Một tụ điện $10\mu F$ nối từ RST xuống GND.
 - Một điện trở $10k\Omega$ kéo lên VCC (5V).
 - Một nút nhấn (SW-PB) nối song song với tụ, cho phép người dùng reset thủ công khi nhấn.

Nguyên lý hoạt động:

- Khi cấp nguồn, tụ C chưa nạp \rightarrow RST ở mức cao (1) \rightarrow Vi điều khiển reset.
- Sau đó, tụ nạp đầy \rightarrow RST trở về mức thấp (0) \rightarrow Vi điều khiển bắt đầu chạy chương trình.
- Khi nhấn nút, tụ bị xả nhanh \rightarrow RST lại lên 1 \rightarrow Vi điều khiển reset lại.

3.6.3. Mạch dao động

Để tạo dao động hệ thống, vi điều khiển cần xung clock ổn định. Mạch sử dụng thạch anh 24MHz nối vào hai chân dao động của vi điều khiển:

- **Chân XTAL1 (chân 19)** nối một đầu thạch anh.
- **Chân XTAL2 (chân 18)** nối đầu còn lại.
- **Hai tụ gồm $33pF$** lần lượt nối từ hai đầu thạch anh xuống mass (GND) nhằm ổn định dao động.

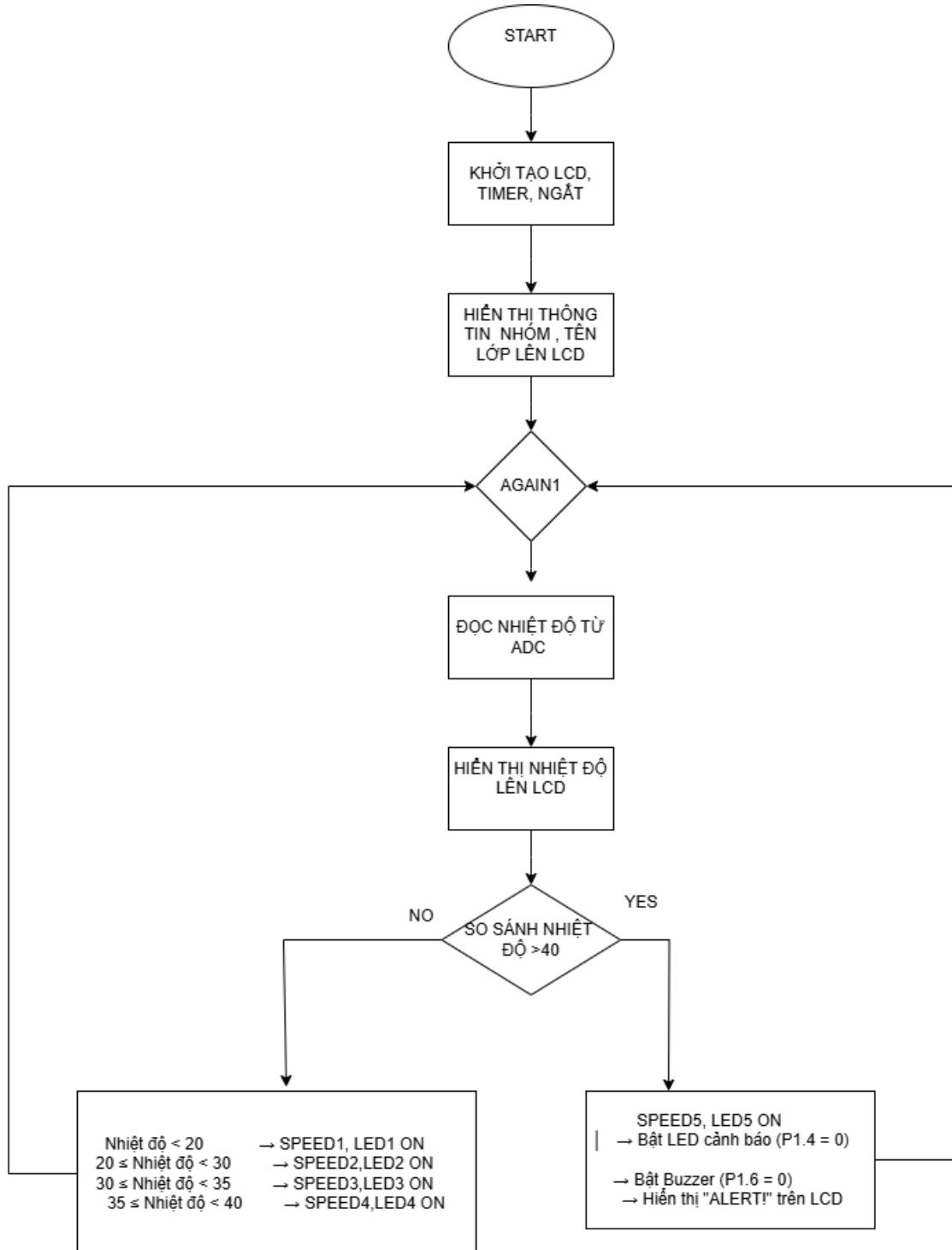
Lý do chọn thạch anh 24 MHz:

- Vi điều khiển hoạt động nhanh, đáp ứng tốt cho xử lý ADC, UART, điều khiển LED, LCD và quạt.

- Xung nhịp lớn giúp Timer, UART hoạt động chính xác nhưng đòi hỏi cấu hình chuẩn xác hơn trong chương trình.

3.7. Thiết kế phần mềm với code Assembly

3.7.1. Lưu đồ giải thuật



Hình 3.7.1 Lưu đồ giải thuật

3.7.2 Khởi tạo cho vi xử lý

Chức năng:

Chuẩn bị cho vi điều khiển hoạt động bằng cách thiết lập trạng thái ban đầu cho các chân vào/ra (I/O), nhằm đảm bảo các thiết bị ngoại vi như LCD, quạt, LED, ADC hoạt động đúng cách ngay khi hệ thống khởi động.

Hoạt động:

- Port 0 (P0):
 - Được cấu hình là ngõ vào để nhận dữ liệu từ ADC0804.
 - Dữ liệu này là giá trị nhiệt độ số hóa từ cảm biến LM35.
- P1.0 – P1.3 (LED báo tốc độ quạt):
 - Mỗi LED biểu thị một mức tốc độ quạt: thấp, trung bình, cao, rất cao.
 - Ban đầu được thiết lập ở mức cao (logic 1) → LED tắt → quạt chưa hoạt động.
 - Khi có tốc độ tương ứng, các LED này sẽ sáng (tùy thiết kế kiểu măc LED: active high hoặc active low).
- P1.4 (Buzzer):
 - Mục đích: báo hiệu khi hệ thống bắt đầu có nhiệt độ cao
 - Thiết lập mức thấp (logic 0) ngay từ đầu
- P1.5 (Nối vào Quạt):
 - Thiết lập ban đầu là mức thấp (logic 0) để quạt chưa hoạt động
- P1.6 (LED báo hệ thống hoạt động đang quá nhiệt độ):
 - LED này được bật sáng để thông báo nhiệt độ đang cao
 - Thiết lập mức cao ngay từ đầu

3.7.3 – Khởi tạo và cấu hình LCD

Chức năng:

Cấu hình LCD 16x2 để hiển thị các thông tin cần thiết như: nhiệt độ, trạng thái quạt, cảnh báo...

Hoạt động:

- Gửi các lệnh chuẩn cho LCD để:
 - Thiết lập giao tiếp 8-bit, hiển thị 2 dòng, font chữ 5x7.
 - Bật hiển thị màn hình và con trỏ.
 - Xóa toàn bộ màn hình.
 - Đặt vị trí con trỏ về đầu dòng.

3.7.4. – Hàm COMMAND

Chức năng:

Gửi lệnh điều khiển (xóa màn hình, di chuyển con trỏ, chọn chế độ hiển thị...) đến LCD.

Hoạt động:

- Lệnh cần gửi được lưu vào **thanh ghi A**.
- A được xuất ra **Port 2** (nối với chân dữ liệu của LCD).
- Chân điều khiển:
 - **RS = 0**: Chế độ lệnh.
 - **RW = 0**: Ghi dữ liệu.
 - **EN = 1 → 0**: Tạo xung để LCD chấp nhận lệnh.\

3.7.5. – Hàm DELAY

Chức năng:

- Tạo khoảng dừng ngắn (vài mili giây) để chờ LCD xử lý lệnh hoặc dữ liệu.

Hoạt động:

- Sử dụng vòng lặp đếm lùi (ví dụ: decrement thanh ghi Rx) để giữ CPU bận trong thời gian nhất định.

3.7.6 – Cấu hình Timer0

Chức năng:

Dùng bộ đếm Timer0 ở chế độ 16-bit để tạo ra các ngắt định kỳ, phục vụ việc kiểm tra nhiệt độ định kỳ, cảnh báo, hoặc nhấp nháy LED.

Hoạt động:

- Đặt Timer0 ở **Mode 1** (16-bit).
- Nạp giá trị ban đầu vào **TH0, TL0** để tạo chu kỳ thời gian mong muốn.
- Khi tràn (overflow), **gây ngắt**, CPU nhảy vào **trình phục vụ ngắt**.

3.7.7 – Bật ngắt toàn cục và Timer0

Chức năng:

Cho phép hệ thống sử dụng ngắt Timer để thực hiện công việc định kỳ (ví dụ: nhấp nháy LED, kiểm tra cảnh báo...).

Hoạt động:

- **EA = 1**: Bật **ngắt toàn cục** (global interrupt).
- **ET0 = 1**: Bật ngắt riêng cho **Timer0**.
- **TR0 = 1**: Kích hoạt **Timer0** bắt đầu đếm thời gian.

3.7.8 – Hiển thị mã môn học và lớp

Chức năng:

Hiển thị thông tin định danh “Do An VXL_VDK” “MA: CE103.P23” trên dòng 2 của LCD.

Hoạt động:

- Đặt con trỏ tới dòng 2 (gửi lệnh tương ứng cho LCD).
- Gọi **hàm LCDWRITE** nhiều lần để gửi từng ký tự trong chuỗi.

Hàm LCDWRITE

Chức năng:

Gửi 1 ký tự bất kỳ để hiển thị lên LCD.

Hoạt động:

- Ký tự được đưa vào **A**.

- Gửi nội dung của A ra **Port 2**.
- Đặt chân điều khiển:
 - **RS = 1**: Gửi dữ liệu.
 - **RW = 0**: Ghi.
 - **EN = 1 → 0**: Xung để LCD nhận dữ liệu.

3.7.9 – Đọc dữ liệu nhiệt độ từ cảm biến

Chức năng:

Đọc giá trị nhiệt độ được chuyển đổi từ analog sang digital bởi **ADC0804**.

Hoạt động:

- Gửi xung điều khiển để bắt đầu chuyển đổi (START CONVERSION).
- Chờ chân **INTR** của ADC xuống mức thấp (hoàn tất chuyển đổi).
- Kích hoạt chân **RD** → dữ liệu nhiệt độ 8 bit được đưa vào **Port 0**.

3.7.10 – Xử lý dữ liệu nhiệt độ

Chức năng:

Dựa trên dữ liệu nhiệt độ vừa đọc, thực hiện:

- Điều khiển tốc độ quạt.
- Hiển thị lên LCD.

Hoạt động:

- Lưu nhiệt độ vào **R4**.
- Gọi hàm **COMPARE** để chọn mức tốc độ quạt phù hợp.
- Gọi hàm **CONVERSION** để tách số nhiệt độ thành 2 ký tự.
- Gọi **LCDWRITETMP** để hiển thị nhiệt độ lên LCD.
- Gọi **DELAY1** để chờ trước lần đọc tiếp theo.

Hàm **DELAY1**

Chức năng:

Tạo khoảng trễ dài hơn DELAY, thường vài trăm mili giây.

Hoạt động:

- Vòng lặp lồng nhau với nhiều biến đếm để giữ CPU bận trong thời gian dài hơn (chống nhấp nháy LCD, tránh đọc ADC quá nhanh).

3.7.11 – So sánh nhiệt độ và điều khiển quạt

Chức năng:

Dựa vào giá trị nhiệt độ đọc được, quyết định mức tốc độ quạt cần thiết hoặc kích hoạt cảnh báo.

Hoạt động:

- So sánh giá trị trong R4 với các ngưỡng:
 - < 20°C → Quạt mức thấp nhất (SPEED 1)
 - 20 – 30°C → Quạt tốc độ thấp (SPEED2)
 - 30 – 34°C → Quạt tốc độ trung bình (SPEED3)
 - 35 – 39°C → Quạt tốc độ cao (SPEED4)
 - 40 – 44°C → Quạt tốc độ cực cao (SPEED5)
 - Gọi hàm ALERT để cảnh báo
 - >= 45°C → Tắt toàn hệ thống, dừng quạt, tắt LED cảnh báo

3.7.12 – Hiển thị nhiệt độ lên LCD

Chức năng:

Hiển thị dòng thông tin như “TEMP: 34C” lên dòng đầu tiên của LCD.

Hiện ở dòng thứ 2: PWM (ví dụ: “PWM: 60%”)

Hoạt động:

- Gọi các hàm hiển thị chuỗi “TEMP:”.
- Gửi 2 ký tự tương ứng với nhiệt độ hiện tại.
- Gửi ký tự “C” (độ C).

3.7.13 – Chuyển nhiệt độ sang dạng ký tự

Chức năng:

Chuyển số nguyên từ ADC (0–99) sang ký tự ASCII để hiển thị LCD.

Hoạt động:

- **Lấy hàng chục:** chia số cho 10.
- **Lấy hàng đơn vị:** số mod 10.
- Mỗi chữ số cộng thêm **0x30 (ASCII ‘0’)** → chuyển sang ký tự.
- Lưu vào thanh ghi dùng trong LCDWRITE.

3.7.14 – Hiển thị cảnh báo "ALERT!"

Chức năng:

Thông báo cho người dùng biết nhiệt độ vượt ngưỡng cho phép.

Hoạt động:

- Đặt con trỏ LCD đến dòng 2.
- Gửi chuỗi “ALERT!”.

3.7.15 – Trình phục vụ ngắt của Timer0

Chức năng:

Xử lý công việc lặp lại mỗi khi có ngắt Timer0, ví dụ: nhấp nháy LED cảnh báo.

Hoạt động:

- Khi có ngắt Timer:
 - Đảo bit P1.5 → điều chế PWM
 - Nạp lại TH0, TL0 để tiếp tục chu kỳ.
- Kết thúc bằng lệnh RETI để trở lại chương trình chính.

CHƯƠNG 4 KẾT QUẢ NGHIÊN CỨU

4.1 Kết quả nghiên cứu

Dưới đây là các trạng thái tiêu biểu trong quá trình mô phỏng

Trạng thái ban đầu

- LCD hiển thị dòng thông tin:

Do An VXL-VDK

MA: CE103.P23

- Các LED báo mức đều sáng.
- Quạt chưa hoạt động (PWM duty = 0%).

Trạng thái khi nhiệt độ tăng dần

Nhiệt độ đầu vào (P0)	PWM Duty (%)	Trạng thái LCD	LED bật	Điện áp Volt điều chỉnh tốc độ quay	Cảnh báo
19	20%	TEMP:19C, PWM:20%	P1.0	~ +0.51	Không
29	40%	TEMP:29C, PWM:40%	P1.1	~ +1.45	Không
34	60%	TEMP:34C, PWM:60%	P1.2	~+2.2	Không
39	80%	TEMP:39C, PWM:80%	P1.3	~+2.8	Không
44	100%	ALERT TEMP:44C, PWM:100%	P1,6	~+3.4	Có
≥45	100% → 0% (ngắt PWM)	TEMP:45C	Tắt	0	Không

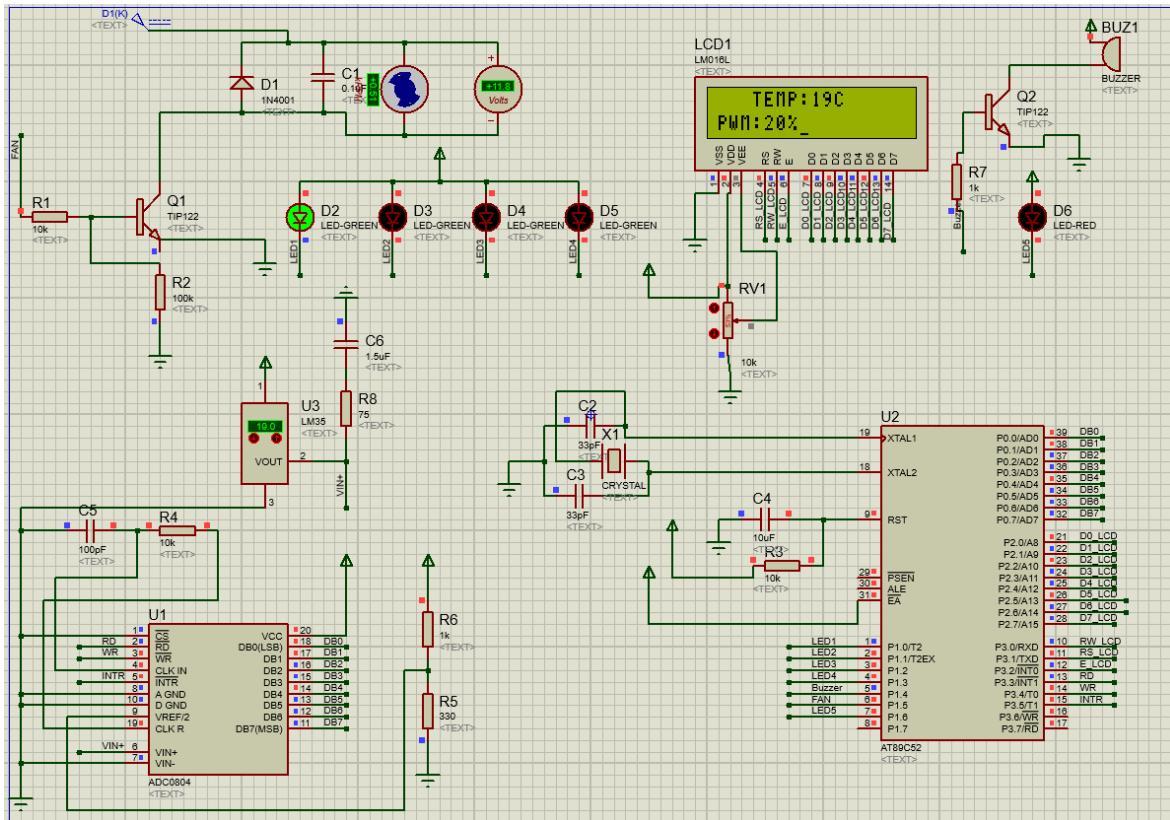
Bảng 4.1 Mô phỏng mạch trong Proteus

- Khi nhiệt độ từ 45, LCD chuyển sang hiển thị "ALERT!", tắt PWM và vô hiệu hóa Timer0.

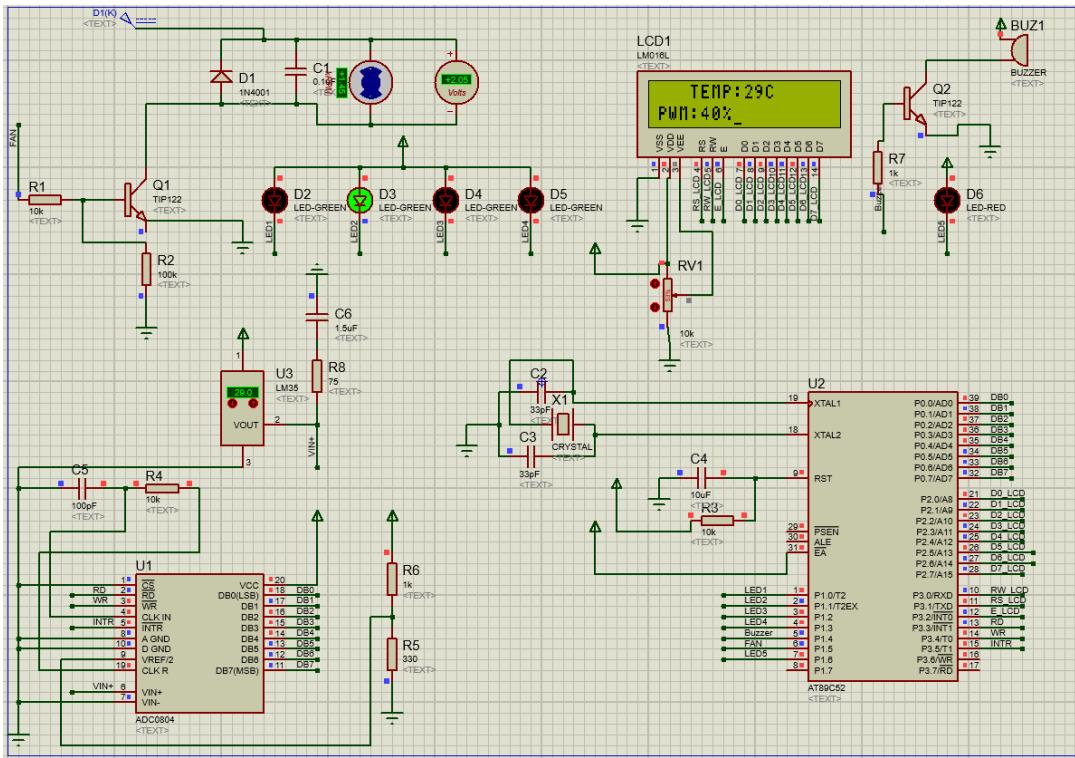
- PWM trên chân **P1.5** hoạt động đúng với chu kỳ và độ rộng xung tương ứng theo biến PWM_DUTY.

Hình ảnh minh họa (chụp từ Proteus):

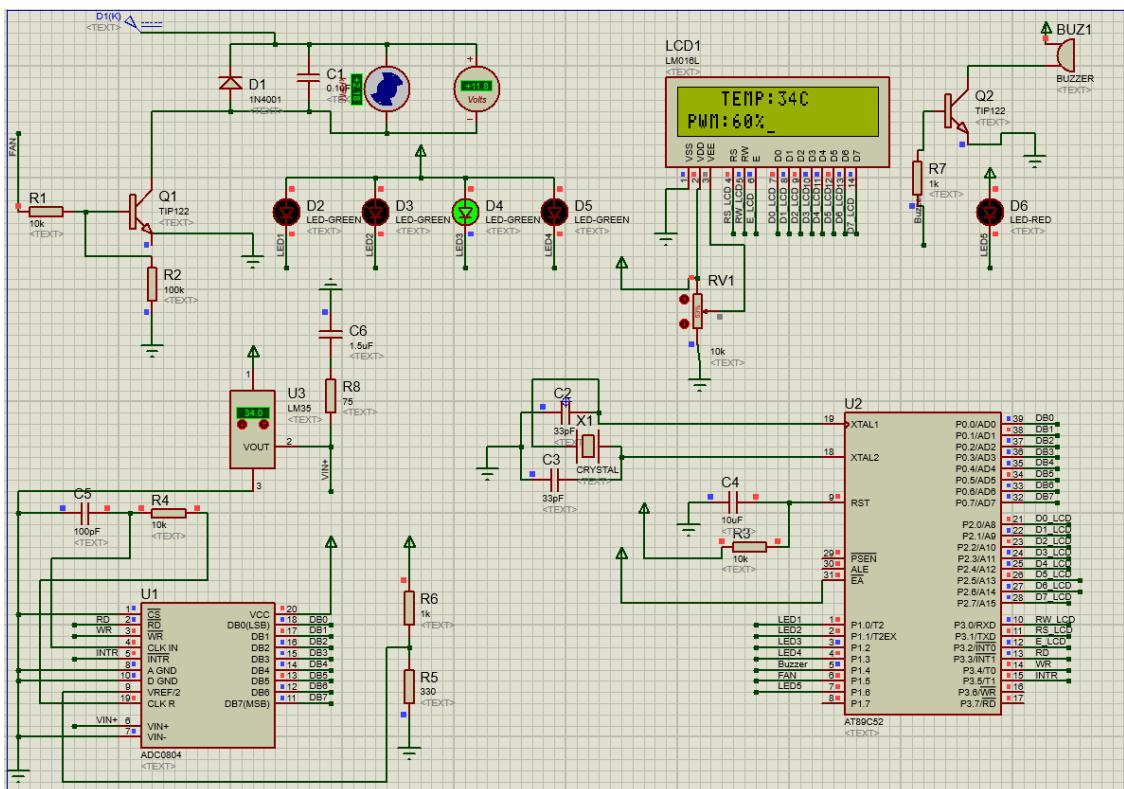
Hình 4.1.1 Khi nhiệt độ là 19 độ C



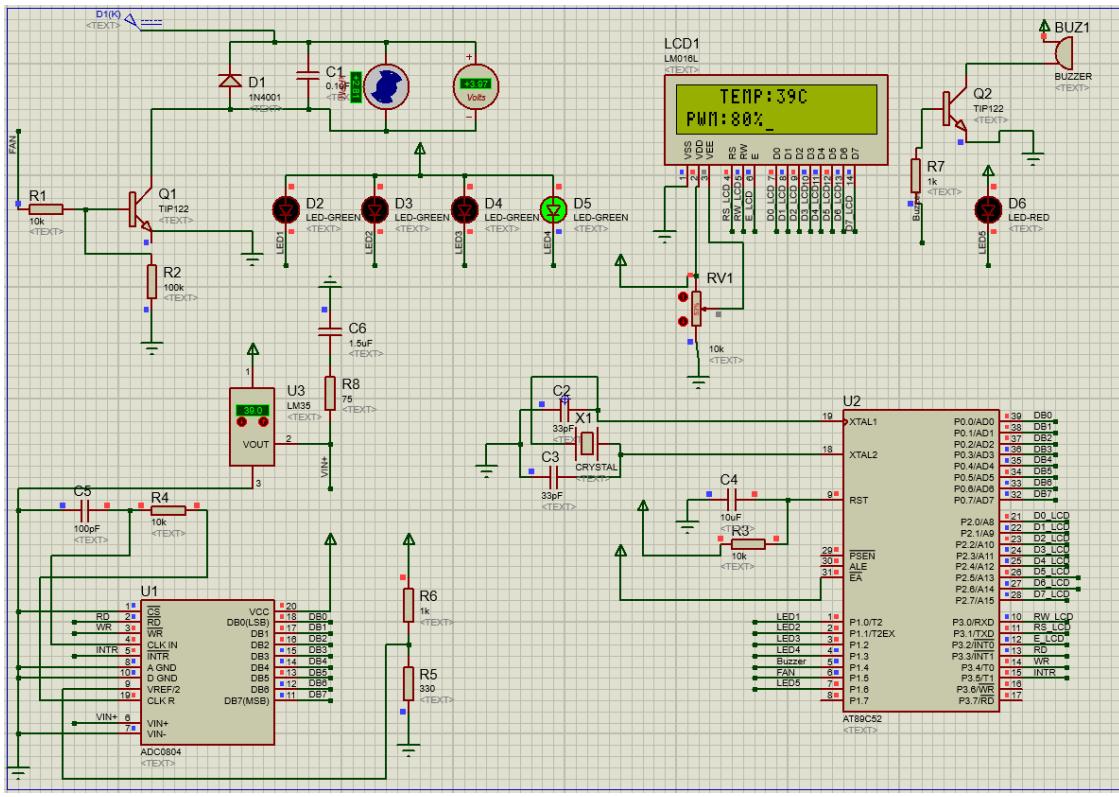
Hình 4.1.2 Khi nhiệt độ là 29 độ C



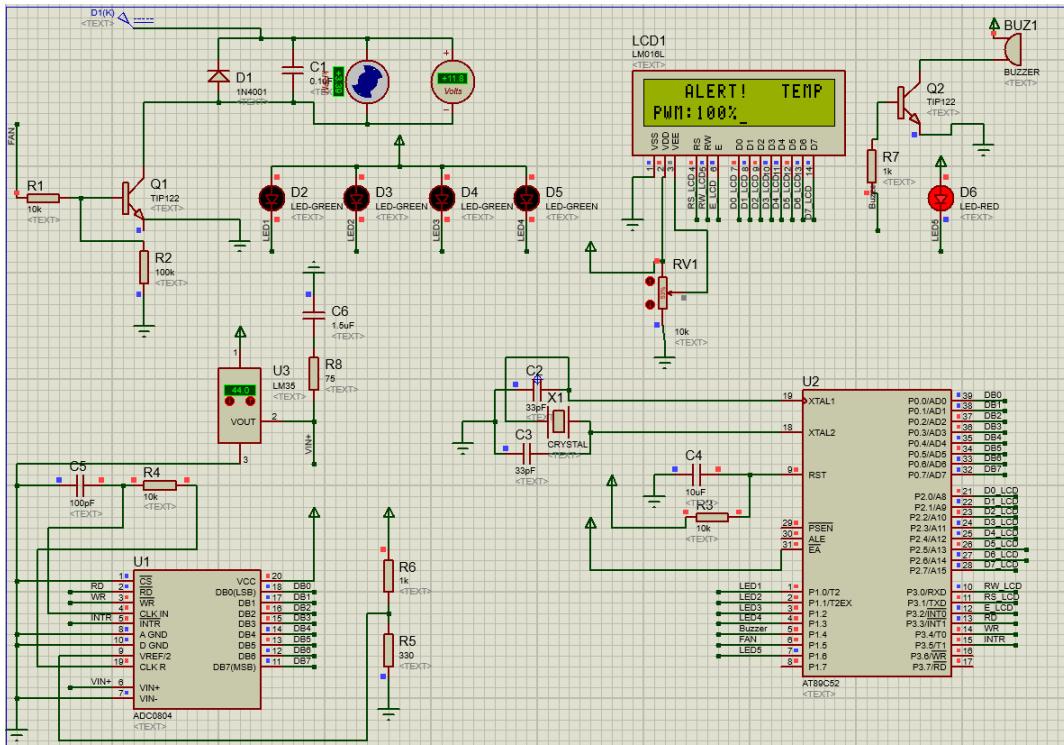
Hình 4.1.3 Khi nhiệt độ là 34 độ C



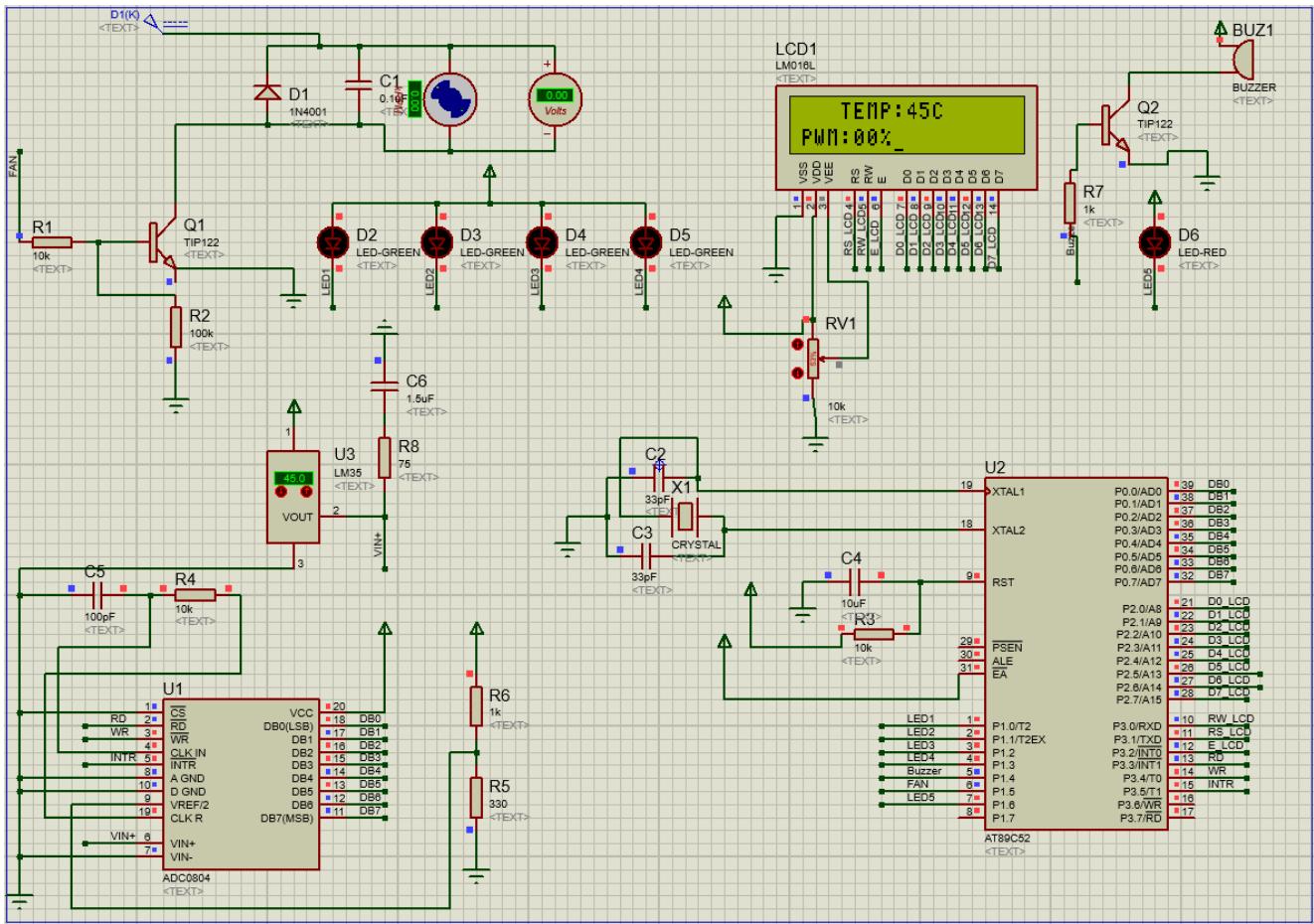
Hình 4.1.4 Khi nhiệt độ là 39 độ C



Hình 4.1.5 Khi nhiệt độ là 44 độ C



Hình 4.1.6 Khi nhiệt độ là 45 độ C(≥ 45 độ C)



Nhận xét kết quả mô phỏng:

1. Tính ổn định và chính xác của chương trình

- Chương trình thực thi ổn định, không xuất hiện treo hệ thống hoặc lỗi logic trong suốt quá trình mô phỏng.
- Giá trị nhiệt độ từ cảm biến LM35 được cập nhật liên tục và chính xác, phản ánh đúng sự thay đổi của nhiệt độ môi trường.
- Việc tính toán và chia mức tốc độ PWM dựa trên giá trị nhiệt độ được thực hiện đúng logic như thiết kế ban đầu (ví dụ: dưới 30°C = quạt ngừng quay, trên 30°C thì PWM tăng dần).

2. Hiệu quả điều chỉnh tốc độ quạt bằng PWM

- Tín hiệu PWM được tạo nhờ Timer 0 và ngắn, đảm bảo chu kỳ ổn định, tỷ lệ duty cycle thay đổi mượt mà theo mức nhiệt.
- Dòng PWM được xuất ra chân P1.5 cho thấy sự thay đổi rõ rệt về độ rộng xung khi nhiệt độ thay đổi.
- Quan sát tín hiệu PWM bằng Oscilloscope trên Proteus cho thấy:
 - Chu kỳ PWM không bị méo hoặc nhiễu.
 - Duty cycle thay đổi đúng theo mức nhiệt đã định.
- Quạt mô phỏng phản ứng hợp lý theo tín hiệu PWM: quay nhanh hơn khi nhiệt độ cao hơn.

3. Hệ thống cảnh báo vượt ngưỡng

- Khi nhiệt độ vượt ngưỡng cảnh báo (ví dụ 40°C hoặc 50°C), chương trình hiển thị dòng cảnh báo trên LCD một cách kịp thời và rõ ràng.
- Cảnh báo không bị lặp lại liên tục (do có điều kiện kiểm tra để tránh ghi đè LCD).
- Nếu có điều khiển LED hoặc còi báo (tùy mô hình), thì cũng hoạt động đúng thời điểm.

4. Giao diện LCD

- LCD hiển thị đầy đủ các thông tin: dòng tiêu đề, giá trị nhiệt độ hiện tại, trạng thái quạt (Tắt, Mức 1, Mức 2...).
- Font và vị trí hiển thị được sắp xếp hợp lý, dễ đọc trong mô phỏng.
- Cập nhật LCD ổn định, không bị nhấp nháy, không sai dữ liệu.
- Giao diện LCD thân thiện với người dùng và dễ theo dõi tình trạng hệ thống trong quá trình chạy.

5. Mạch mô phỏng và phần cứng

- Mạch mô phỏng Proteus hoạt động ổn định, không xuất hiện xung đột chân hoặc lỗi kết nối.
- Tín hiệu từ vi điều khiển qua TIP122 được khuếch đại đúng cách để điều khiển quạt mô phỏng.

- Các linh kiện phụ trợ như: điện trở, tụ, diode chống ngược... góp phần giúp mạch hoạt động ổn định, không bị nhiễu trong suốt quá trình mô phỏng.
- ⇒ Mô phỏng đã chứng minh hệ thống điều khiển quạt tự động hoạt động đúng như thiết kế. Các tính năng từ đo nhiệt độ, xử lý tín hiệu PWM, hiển thị LCD cho tới cảnh báo đều thể hiện tốt trong môi trường mô phỏng Proteus. Hệ thống có thể triển khai thực tế với hiệu chỉnh nhỏ về phần cứng.

4.2 Đánh giá hiệu năng

Hệ thống điều khiển quạt tự động theo nhiệt độ sử dụng vi điều khiển AT89C52 đã hoàn thành với đầy đủ các chức năng cơ bản như: đo nhiệt độ môi trường, hiển thị thông tin trên màn hình LCD, điều khiển tốc độ quạt theo từng mức nhiệt độ cụ thể và cảnh báo khi nhiệt độ vượt ngưỡng an toàn. Qua quá trình kiểm tra và vận hành thực tế, hệ thống cho thấy hoạt động ổn định, xử lý nhanh nhạy với sự thay đổi của môi trường xung quanh.

Cấu trúc hoạt động của hệ thống như sau:

- **Cảm biến nhiệt độ LM35** đo nhiệt độ môi trường và xuất tín hiệu analog.
- **Bộ chuyển đổi ADC0804** nhận tín hiệu analog và chuyển thành dữ liệu số gửi về vi điều khiển.
- **Vi điều khiển AT89C52** xử lý dữ liệu và đưa ra tín hiệu điều khiển quạt thông qua các chân xuất của cổng P1.
- **Màn hình LCD 16x2** hiển thị thông tin bao gồm nhiệt độ, tên sinh viên và mã lớp.
- **Còi cảnh báo** được kích hoạt khi nhiệt độ vượt quá 40°C và dưới 45°C.

Quạt được điều khiển theo các mức nhiệt như sau:

- **Dưới 20°C:** Quạt tắt hoàn toàn.
- **Từ 20°C đến 29°C:** Quạt hoạt động ở mức tốc độ thấp.
- **Từ 30°C đến 34°C:** Quạt chạy ở mức tốc độ trung bình.
- **Từ 35°C đến 39°C:** Quạt chạy ở mức trung bình cao.
- **Từ 40°C đến 44°C:** Quạt chạy ở mức tốc độ cao cùng còi báo động.
- **Từ 45°C trở lên:** Quạt và còi báo động sẽ tắt

Ngoài ra, hệ thống còn sử dụng bộ định thời (Timer0) để tạo hiệu ứng chớp tắt cho còi hoặc LED cảnh báo, làm tăng tính trực quan khi có tình huống khẩn cấp.

Tổng kết: Hệ thống hoạt động tốt, ổn định, có phản hồi nhanh, và đã được tối ưu hóa để phù hợp với tài nguyên hạn chế của dòng vi điều khiển AT89S52.

4.3 Ưu và nhược điểm của hệ thống

Ưu điểm

1. Tự động hóa hoàn toàn:

Hệ thống hoạt động hoàn toàn tự động, không cần sự can thiệp thủ công, giúp nâng cao tính tiện lợi và tiết kiệm thời gian.

2. Hiển thị thông tin rõ ràng:

Màn hình LCD 16x2 giúp người dùng dễ dàng theo dõi nhiệt độ và trạng thái quạt/tín hiệu cảnh báo trong thời gian thực.

3. Phản hồi nhanh và chính xác:

Vì điều khiển xử lý tín hiệu nhanh chóng khi có sự thay đổi nhiệt độ, đảm bảo quạt vận hành tức thời và hiệu quả.

4. Tích hợp cảnh báo nhiệt độ cao:

Hệ thống có còi cảnh báo khi nhiệt độ vượt quá ngưỡng cho phép ($\geq 45^{\circ}\text{C}$), giúp đảm bảo an toàn trong những tình huống khẩn cấp.

5. Chi phí thấp và dễ triển khai:

Sử dụng các linh kiện thông dụng như LM35, ADC0804, LCD 16x2, giúp sinh viên dễ tìm linh kiện, chi phí rẻ, phù hợp cho giáo dục và thực hành.

6. Cấu trúc mạch đơn giản:

Thiết kế mạch điện đơn giản, dễ lắp ráp, thuận tiện trong việc kiểm tra, sửa lỗi và nâng cấp.

Nhược điểm

1. Độ chính xác giới hạn:

Cảm biến LM35 và bộ chuyển đổi ADC0804 có sai số nhất định, chưa đạt độ chính xác cao như các cảm biến kỹ thuật số hiện đại.

2. Chưa có khả năng lưu trữ và giám sát từ xa:

Hệ thống hiện tại không lưu trữ dữ liệu lịch sử nhiệt độ và không thể điều khiển hoặc theo dõi từ xa qua mạng.

3. **Khả năng mở rộng còn hạn chế:**

Sử dụng vi điều khiển AT89C52 có tài nguyên hạn chế (RAM/ROM/ngoại vi), gây khó khăn khi muốn thêm nhiều tính năng mới (như kết nối Internet, lưu trữ dữ liệu...).

4. **Chưa có giao diện người dùng nâng cao:**

LCD 16x2 chỉ hiển thị văn bản đơn giản, chưa thể hiển thị đồ họa, biểu đồ nhiệt độ hay giao diện thân thiện cho người dùng.

4.4. Hướng mở rộng cho hệ thống

Để nâng cao tính năng và phạm vi ứng dụng thực tiễn của hệ thống, có thể mở rộng theo các hướng sau:

a. Về phần cứng:

- Thay LM35 và ADC0804 bằng cảm biến nhiệt độ kỹ thuật số như DS18B20 hoặc DHT11 để đơn giản hóa giao tiếp và tăng độ chính xác.
- Ứng dụng điều khiển tốc độ quạt bằng PWM thay vì bật/tắt mức cố định, giúp quạt chạy êm và tiết kiệm điện năng hơn.
- Tích hợp cảm biến độ ẩm hoặc khí gas để xây dựng hệ thống giám sát môi trường thông minh.
- Trang bị pin dự phòng (UPS) để hệ thống tiếp tục hoạt động khi mất điện – phù hợp với các ứng dụng giám sát nhiệt độ trong tủ điện hoặc server.

b. Về phần mềm:

- Nâng cấp giao diện hiển thị bằng màn hình OLED hoặc TFT để thể hiện biểu đồ nhiệt độ và trạng thái thiết bị.
- Lưu trữ dữ liệu nhiệt độ theo thời gian vào EEPROM hoặc thẻ nhớ SD phục vụ phân tích và giám sát lâu dài.
- Tích hợp kết nối không dây (Bluetooth/Wi-Fi với ESP8266 hoặc ESP32) để người dùng có thể theo dõi và điều khiển hệ thống từ xa.
- Cấu hình ngưỡng nhiệt độ cảnh báo linh hoạt bằng cách cho phép người dùng thiết lập thông qua các nút nhấn hoặc giao tiếp UART.

c. Về ứng dụng thực tế:

- Điều khiển nhiệt độ phòng làm việc, phòng ngủ hoặc phòng máy tính một cách thông minh và tự động.
- Cảnh báo nhiệt độ cao trong tủ điện công nghiệp nhằm bảo vệ thiết bị khỏi quá nhiệt.
- Giải pháp làm mát và cảnh báo trong hệ thống nhà kính thông minh, tích hợp thêm khả năng tưới tiêu và giám sát độ ẩm.

4.5. Kết luận

Đồ án "Hệ thống điều khiển quạt tự động theo nhiệt độ sử dụng vi điều khiển AT89C52" đã hoàn thành đúng mục tiêu đề ra: giám sát và điều khiển hoạt động của quạt dựa trên nhiệt độ môi trường. Hệ thống vận hành ổn định, hiển thị nhiệt độ chính xác trên màn hình LCD và phản hồi linh hoạt qua các mức điều khiển quạt, đồng thời có tích hợp cảnh báo khi nhiệt độ vượt ngưỡng an toàn.

Qua quá trình thiết kế và lập trình hệ thống, nhóm thực hiện đã hiểu rõ hơn về cách hoạt động của cảm biến nhiệt độ, bộ chuyển đổi ADC, giao tiếp LCD, cũng như việc sử dụng ngắt và timer trong vi điều khiển 8051. Đây là nền tảng quan trọng để phát triển các ứng dụng điều khiển tự động khác trong thực tế.

Tuy nhiên, hệ thống vẫn còn một số hạn chế nhất định về độ chính xác, khả năng mở rộng và tính linh hoạt trong cài đặt. Những điểm này sẽ là hướng đi tiếp theo để nâng cấp và phát triển hệ thống ngày càng hoàn thiện hơn, chẳng hạn như ứng dụng PWM để điều chỉnh tốc độ quạt mượt mà hơn, hoặc tích hợp IoT để theo dõi nhiệt độ từ xa.

4.6. Video mô phỏng

Link video:

https://drive.google.com/file/d/1CVDUOGL6WeD8fMp0zAjjh0TgNzn2jI4j/view?usp=drive_link

CHƯƠNG 5 SOURCE CODE VÀ TÀI LIỆU THAM KHẢO

5.1. Source code của mạch

Link: https://github.com/TPhuong24/CE103.P23_TECHSTORM/tree/main

5.2. Tài liệu tham khảo

- [1] Mazidi, Muhammad Ali, Janice Gillispie Mazidi, and Rolin D. McKinlay. *The 8051 Microcontroller and Embedded Systems: Using Assembly and C.* Pearson Education, 2006.
- [2] Ayala, Kenneth J. *The 8051 Microcontroller.* Cengage Learning, 2004.
- [3] Nguyễn Đức Thuận. *Vi điều khiển 8051 và ứng dụng.* NXB Khoa học và Kỹ thuật, 2010.
- [4] Trần Văn Tín. *Vi điều khiển – Lý thuyết và thực hành với 8051.* NXB Đại học Quốc gia TP.HCM, 2015.
- [5] Atmel Corporation. *AT89S52 8-bit Microcontroller with 8K Bytes In-System Programmable Flash.* Atmel Datasheet, 2015. [Online]. Available: <https://www.microchip.com>
- [6] Atmel Corporation. *AT89C52 8-bit Microcontroller with 8K Bytes Flash.* Atmel Datasheet, 2002. [Online]. Available: <https://www.microchip.com>